A neurofuzzy algorithm for learning from complex granules

Bruno Apolloni¹ · Simone Bassis¹ · Jacopo Rota¹ · Gian Luca Galliani¹ · Matteo Gioia¹ · Luca Ferrari¹

Received: 23 July 2015 / Accepted: 15 February 2016 / Published online: 4 April 2016
© Euratom: Università degli studi Milano; © European Union 2016

Abstract In the context of complex granule computations within the Interactive Granular Computing (IGC) paradigm we frame a cognitive task where user perceptions of the suitability of a good are in relation to the parameters of the device producing it, all within a learning loop aimed at continuously improving those perceptions. We achieve this goal by extending the Fuzzy Inference System (FIS) paradigm to contexts where variables reckoning the user perceptions live in a non-metric space, hence neither users nor the learning algorithm have access to their *true* value. Namely, receiving in input a set of both crisp and fuzzy variables (respectively, from the *hard_suit* and the *soft_suit* of the *c*-granule to account for user and device logs), the inference system is asked to compute via the *link_suit* a set of crisp parameters satisfying some fuzzy evaluations stated by the user. A further complication is that the outputs are evaluated exactly in terms of the *true* unknown values

held by the fuzzy attributes, which in turn must be inferred by the system. The whole work arose from everyday life problems faced by the European Project Social&Smart with the aim of optimally regulating household appliance runs. It represents a special instance of Interactive Rough Granular Computing (IRGC) that we face with a two-phase procedure that is reminiscent of the *distal learning* in neurocontrol. A web service is available where the reader may check the efficiency of the assessed procedure.

Keywords Fuzzy inference systems · Interactive granular computing · Complex granules · Distal learning · Two-phase learning

1 Introduction

Within the wide spectrum of theoretical, methodological and technological aspects covered by the Granular Computing framework, let us focus on a relatively new one rooted on the two interwoven concepts of Interactive Granular Computing [IGC (Pedrycz and Chen 2015)] and Complex Information Granule [*c*-granule, Skowron and Jankowski (2015)]. Both arise from the need of putting in relation an internal setting made of information granules with the physical world representing the external environmental counterpart that those granules affect. In the age of user experience (Amershi et al. 2014) these relations have great relevance in many operational situations when we want to optimize the person's perception of pleasantness about using a particular device. A way of dealing with the two contexts passes through the definition of complex information granules (*c*-granules), which are usually related to interplaying agents that face the physical environment through abstractly defined windowed hunks (Heller

This work has been supported by the European Project FP7 317947 Social&Smart.

✉ Bruno Apolloni
apolloni@di.unimi.it
Simone Bassis
bassis@di.unimi.it
Jacopo Rota
jacopo.rota@studenti.unimi.it
Gian Luca Galliani
galliani@di.unimi.it
Matteo Gioia
gioia@di.unimi.it
Luca Ferrari
ferrari@di.unimi.it

¹ Dipartimento di Informatica, Università degli Studi di Milano, Via Comelico 39/41, 20135 Milan, Italy

1990). This definition takes place in terms of the three components: (1) the *soft_suit*, recording perceived properties of hunks and their interactions; (2) the *hard_suit*, responsible for the response of the environment to the agents' interactions in terms of physical variables, and (3) the *link_suit*, piping interactions between agent and environment, allowing the former to perform sensory measurement and/or actions on the physical objects. The granule interactions in the physical and mental worlds are generally exploited among agents that are mainly directed to the social evolution of huge communities of them, addressing the fundamental issues in Wisdom Technology (Jankowski and Skowron 2013), and are implemented in terms of ontology updating (Clark et al. 2013; Skowron et al. 2012), usually under the framework of Interactive Rough Granular Computing [IRGC (Skowron and Szczuka 2010; Kacprzyk and Pedrycz 2015)]. Rather, in this paper we focus on the interactions inside the single *c*-granule, and rule them in terms of a peculiar Fuzzy Inference System [FIS (Provotar et al. 2013)]. The common ground between the two perspectives lies in the *adaptive judgment* about interactive computations on *c*-granules, which allows an inductive and rational evaluation of user-system interactions (Jankowski et al. 2013).

Since their introduction by Zadeh (1965), fuzzy sets have been intended as a rigorous way of dealing with non-formalized knowledge falling in the sphere of the experience and intuition of the designers. In place of *explaining why*, membership functions comprise a clear way of *describing the granularity* of the information owned by them. Within the Granular Computing framework, fuzzy sets constitute the key ingredients of both the theory of approximate reasoning (Zadeh 1979), a powerful framework for reasoning in the face of imprecise and uncertain information, and its operational counterpart known as fuzzy rule systems as well (Pedrycz 2013).

In this paper, we use fuzzy rules to manage the intra-granule interactions, where the two main contributions we provide focus on stretching the Granular Information features to cope with the specific *soft_suit* of our *c*-granule, and on devising the Granular Constructs to deal with them within the *link_suit*, respectively. Namely, we consider fuzzy sets inside a universe of discourse that is hidden to the user and then offer a method for both identifying and controlling fuzzy rule systems based on this kind of fuzzy attributes. We call this method *learning from nowhere*, not to diminish its value, rather to remark on the extremely poor a priori information a Granular Construct may exploit during a training session, and extend the *distal learning* construct from the neural network to the fuzzy set framework to get rid of this drawback. We note that hidden universes of discourse and our *c*-granule constitute a realistic scenario that we may meet very frequently in our

everyday life. In fact, we faced and resolved it in our own the European project SandS,¹ which deals with the conduction of common household appliances. We devote a great part of the paper to the setup and conduction of numerical experiments, due to the relevance of this challenging problem and to the consequent casting out nines it represents for a concrete suitability of some Granular Constructs.

Fuzzy rule systems represent the favorite tool for solving system control problems when fuzziness affects the description of their dynamics (Miu and Hayashi 2000). Even though different interpretations of logical connectives, fuzzy implication and compositional operators lead to a variety of well-known inference mechanisms ruling the semantics of a fuzzy rule-based system (consider for instance the Mamdani, Tsukamoto, Sugeno and Larsen reasoning models), its general expression may be synthesized in the following one (Fullèr 1998):

$$\begin{array}{ll} \text{if } x_1 \text{ is } A_{11} \text{ AND } \dots \text{ AND } x_n \text{ is } A_{1n} \text{ then } o \text{ is } B_1 \\ \text{if } x_1 \text{ is } A_{21} \text{ AND } \dots \text{ AND } x_n \text{ is } A_{2n} \text{ then } o \text{ is } B_2 \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ \text{if } x_1 \text{ is } A_{k1} \text{ AND } \dots \text{ AND } x_n \text{ is } A_{kn} \text{ then } o \text{ is } B_k \end{array} \quad (1)$$

where A_{ij} and B_i , for all $i = 1, \dots, k$ (k —the number of rules), $j = 1, \dots, n$ (n —the number of conditions), are fuzzy sets defined in the corresponding input and output spaces; x_i and o are input and output variables (possibly linguistic) corresponding to, respectively, the i -th condition and conclusion. In the above system, the premise of the i -th rule (prefixed by **if**) maps into its firing strength $w_i(\mathbf{x}) = \bigwedge_{j=1}^n \mu_{A_{ij}}(x_j)$, where $\mu_A(x)$ denotes the membership grade of x to the fuzzy set A . In turn, together with the consequent B_i (prefixed by **then**), firing strength $w_i(\mathbf{x})$ contributes to the individual rule output $B'_i = w_i(\mathbf{x}) \rightarrow \mu_{B_i}(o)$ and, through a proper aggregation operator, to the overall system output, computed as $\bigvee_{i=1}^k B'_i$. The various instances of fuzzy inference schemes arise then from the different interpretations given to the above connectives and operators, as well as from the adopted shapes of membership functions (Fullèr 1998).

Within the mentioned project SandS, where we aimed at optimally and adaptively ruling in remote the microcontrollers of household appliances, we faced peculiar fuzzy rules which, re a bread maker for example, sound as follows:

- **if** the loaf is slightly soft **AND** soggy **then** increase baking time
- **if** the loaf is very crusty **AND** crunchy **then** decrease baking time

¹ <http://www.sands-project.eu>.

Fig. 1 A learning from c -granules instance. In the picture, x_1 and x_2 refer, respectively, to the fuzzy variables crustiness and moistness; the fuzzy sets A_{11} and A_{12} refer to the linguistic variables “slightly soft” and “soggy” (i.e. “poorly crusty” and “very moist”), while A_{21} and A_{22} to “very crusty” and “crunchy” (i.e. “not much moist”), respectively; the output o_j of each Sugeno function describes the numeric contribution of the j -th rule to the baking time. All contributions are gathered in the output of the system through (2)

While in the first instance a Mamdani reasoning model (Mamdani 1977) might seem the most appropriate schema to frame the above rules—mainly due to its widespread acceptance and intuitiveness which makes it well suited to human input, a more in-depth investigation led us to prefer the Sugeno reasoning model (Takagi and Sugeno 1985). Here, the consequents are crisp variables to be computed through a weighted mixture of functions depending directly on the input variables; in formula:

$$o = \sum_{i=1}^k \bar{w}_i f_i(\mathbf{x}, \mathbf{s}) = \frac{\sum_{i=1}^k w_i f_i(\mathbf{x}, \mathbf{s})}{\sum_{i=1}^k w_i} \quad (2)$$

where f_i defines the activation function (for short *Sugeno function*) of the i -rule, whose shape, arguments and free parameters $\mathbf{s}$ depend on the chosen model, and w_i denotes the satisfaction degree of the premise of the rule, i.e. its firing strength. Finally, $\mathbf{x}$ and o refer, respectively, to the crisp input variables and the overall output of the system.

Our preference for this model was mainly motivated by the possibility of injecting prior knowledge in the ruleset, specifically in terms of rules and variables identification, and Sugeno function definition as well, thanks to specific competences and expertises held by household manufacturers and expert practitioners in the field. For greater clarification, returning to the two aforementioned rules, let's consider the pictorial system schema shown in Fig. 1. Recent studies in the bread-making process (Mondal and Datta 2008) have taught us that to have, for instance, a crunchy crust, we have to prolong the baking time and

lower the baking temperature but not intervene directly in the leavening process. This information allows us not only to select a useful subset of rules fine for meeting user needs, but even to identify the shape and possibly some parameters of the involved Sugeno functions.

Of course, this choice is not free from complications. The first drawback, which in turn shows up in the distinguishing features of our fuzzy sets—the first main contribution of our paper—is linked to the fact that some input variables (such as the above crustiness and moistness) live in a non-metric space, in the sense that the user is not able to attribute a specific value to them. In fact, when the user says that “the loaf is very crusty”, even if s/he distinguishes different kinds of crustiness falling within the same quantifier “very crusty”, s/he has no reference metric to identify its specific value. Bread crustiness is a subjective attribute not simply as it may vary on a per user basis; rather, the same user may attribute to a same crustiness level the term “very crusty” in certain contexts/recipes, and “low crusty” in other ones. Vice versa, the same quantifier “very crusty” may incorporate a wide spectrum of crustiness levels, whose relation is hidden from the user her/himself and dynamically changes according to external non-measurable factors. In other words, crustiness level x_j exists, but its value cannot be objectively set by the user, who in turn limits her/himself to assigning it a quantifier s/he retains suitable during a particular period of her/his life. Nevertheless, to identify a suitable value of the output o in (2) (both to compute the satisfaction degree of the

premise and to instantiate the Sugeno function in the consequent) we must estimate the values of such hidden variables—a problem that per se is not far from the one of identifying the state transition matrix in the Hidden Markov Models (Rabiner and Juang 1986); it does, however, run into special difficulties given the control framework where it is embedded. The second main contribution of this paper consists in overcoming these difficulties.

Actually, the typical taxonomy for Granular Constructs (Apolloni et al. 2008) shown in Fig. 2 considers the two alternatives where the antecedent is either a crisp variable to be framed into a fuzzy set by the interface (first row in the picture), or the fuzzy set as a whole (second row). *Our case is an intermediate one*, where we need the crisp variables both to compute the satisfaction degree of the premise and to instantiate the Sugeno function in the consequent. But we do not know their values. Typical approaches for further stressing the fuzziness of some variables are represented by type 2 fuzzy sets (Mendel and John 2002), where membership degrees are in turn fuzzy sets, and by higher order fuzzy sets (Pedrycz 2013), where the universe of discourse of a fuzzy set is in turn made up of fuzzy sets. Our approach is, rather, reminiscent of the fuzzy set calibration introduced in Pedrycz et al. (1997). In that case a proper non-linear mapping is learned from an original universe of discourse to another one that may prove more suitable for supporting the fuzziness of some attributes. In our case this mapping remains implicit, since we do not know the original universe and aim directly to infer the single elements of it which support the questioned attributes.

The identification of the crisp variables—the core of the *hard_suit* component of our *c*-granule—is further complicated by the special features of the Sugeno functions f_i s in two respects. On the one hand, in the absence of expert knowledge, we dare to use functional forms, such as linear or quadratic ones, to interpret consequents like “increase/decrease” of the baking time. Although a common plague of the Sugeno approach (Cococcioni et al. 2007), it reflects a fortiori the robustness and flexibility of this model, capable of fitting highly non-linear trends in spite of the simple (possibly linear) regressors each rule produces in output. On the other hand, we do not know the relation between the operational parameter (the baking time) and its effect on the crustiness and moistness appreciation provided by the user. This entails a non-trivial identification problem to be faced in a way reminiscent of *distal learning* in neurocontrol (Jordan and Rumelhart 1992) with a two-phase (identification and control) algorithm. In particular, the operational parameters (i.e. baking time and temperature, leavening time, etc.), being directly controlled by the learner, play the role of *proximal* variables in the distal learning framework. Actually, these variables are the only levers the learner can move to modify the bread consistency and quality. Conversely, user judgments (re baking crustiness, for instance) are *distal* variables that the learner controls indirectly through the intermediary of the proximal variables. The only assumption the whole system relies on is that the target values of user judgments be always available; actually, these requirements are always straightforwardly met: independently of the *true* (unknown) level of crustiness, in case the user is satisfied with

Fig. 2 Four fundamental modes of the use of fuzzy models (Apolloni et al. 2008)

the crust consistency, s/he will not request any further adjustment aimed at increasing/decreasing its texture.

To give some hints on the information flow characterizing the proposed FIS together with the interrelate role played by both proximal and distal variables, let's consider a very simplistic system composed of the following single rule:

if baking time is somewhat high AND baking temperature is very high AND the loaf is slightly soft AND soggy
then increase baking time very much AND lower baking temperature a bit

Firstly, both operational crisp parameters (baking time and temperature) and user fuzzy judgments (bread softness and soggianness) are provided in input to the rule, where the former characterize the current status of the system to be controlled (i.e. the bread machine), while the latter represent the user appreciation of the bread produced by the device *in its actual configuration*. Then the rule controls the two operational parameters by invoking two Sugeno functions aimed at estimating how the parameters should be modified to meet user requirements. Proceeding with the lead example, we find that the output of this rule might be “increase the baking time by 12 seconds and lower the temperature by 10 degrees”. The person determining whether or not the proposed settings of the system are optimal, or at least preferable w.r.t. previous ones, is the user who, after having tasted the home-baked bread, is called upon to evaluate its softness and soggianness through two fuzzy judgments. The latter will have a twofold role in the learning process: they represent both the target to be achieved by the system and at the same time its fuzzy input for the next trial. As for the former, the system will use these values to tune the way operational parameters are modified so as to achieve a configuration satisfying the user tastes, corresponding to the barycentric position w.r.t. the two evaluations. As for the latter, the FIS per se may be considered as a dynamic system where learner, user, and indirectly the bread machine, continually interrelate with another until an optimal configuration is reached. So, the input of the system at the next time step will be constituted both by the operational parameters obtained in output at the previous iteration and the same user fuzzy judgments playing the role of target values at the previous time step.

At this point, some remarks are noteworthy.

1. We are free to select the structure of the rules, with the aim of achieving a suitable balance between informativeness of the system and its trainability, where the

former calls for a high number of rich rules; vice versa for the latter. On the one hand, we may subdivide the overall FIS in various parts, called *clusters* henceforth, each one deputed to tune a single operational parameter. On the other hand, while letting the target be constituted by all the available judgments, to reduce the number of rules we may consider as input only the operational parameter to be tuned in the cluster the rule belongs to.

2. The extra complexity w.r.t. a standard FIS is constituted by *learning from nowhere* the value of those variables related to user evaluations. The true values of bread softness and crustiness, while hidden from both FIS and user her/himself, are needed to compute both the rule strength and the output of the Sugeno function. The aim of the whole process is then to learn these values along with the parameters the FIS is composed of, so as to meet user requirements. Obviously, we may bargain with respect to the shape of the membership functions and the non-linearities of the scales of these variables. However, as to the two extremal alternatives: (linear scale (no matter the meaning of the variables) and highly complex membership functions), (unknown scale (we just infer the coordinates of the sample points), elementary membership functions (such as triangular or Gaussian)), implicitly in line with (Pedrycz et al. 1997), we prefer the latter, as a sort of kernel trick (Hofmann et al. 2008) applied to fuzzy rules.
3. To tune the system according to the user judgments, our extension to the fuzzy rules of the two-phase algorithm proposed in the distal learning framework (Jordan and Rumelhart 1992) consists of an initial task where the learner identifies the system by discovering, through a classical supervised learning task, the mapping between operational parameters and user judgments (hence the term *identification* phase). After having identified the system, the learner switches to a *control* phase, aimed at regulating the system so as to satisfy user needs.
4. No assumption is made on the expertise level and competencies of the user. In the lead example, s/he plays the role of bread tester: taking a loaf out of the bread maker, s/he relies only on her/his palate and sense of taste to evaluate the bread's consistency. In turn, the inference system should be able to adapt to the user needs even in case of out-of-the-ordinary comments and judgments. Users should not to be confused with experts, such as appliance manufacturers. Contrary to the former, the latter may directly inject some prior knowledge to the system, by

suggesting, for instance, the most adequate rules or the shape of Sugeno functions, as explained in the next section.

In our paper we implement our c -granule in two scenarios: (1) a case study where, using a graphical tool, we are challenged to beautify a face whose morphological parameters are computed, on the basis of user evaluations based on four criteria, by a FIS trained on a relatively small and noisy dataset; and (2) the actual assessment of the SandS system on a bread maker.

The paper is organized as follows. In Sect. 2 we formalize the training problem and specify the adopted options. In Sect. 3 we describe the two experimental scenarios, framing the latter within the SandS project, while in Sect. 4 we discuss the numerical results. Conclusions are drawn in Sect. 5.

2 The formal framework

Having framed our inference instance in the IGC paradigm in the Sect. 1, we focus in this section on its solution in terms of the training of a FIS that we characterize by:

- n_c crisp variables y_i s, associated to the operational parameters;
- n_f fuzzy variables g_i s, each described by r linguistic quantifiers, associated to the task execution evaluation;
- k rules, grouped in n_c clusters each containing $k_c = r^{n_f+1}$ rules, where each cluster is responsible for the update of the single operational parameter, and each rule consists in principle of $n_f + 1$ antecedents (the n_f fuzzy variables plus the operational parameter the cluster refers to).

Framing the lead example in the above context, Fig. 1 shows two out of the k_c rules constituting the cluster responsible for the baking time, where we have $n_f = 2$ fuzzy variables g_1 and g_2 (loaf crustiness and humidity) but no crisp variable. Actually, according to the system architecture described in the above bullet list, a realistic model, such as the one adopted in our experiments, would include as input even the crisp variable baking time y_1 , after a suitable fuzzification. In the experiments we fixed $r = 3$ as a suitable compromise between richness of the model and manageability in operational terms. For instance, the three fuzzy sets associated to loaf crustiness have been chosen as: *poorly*, *regular*, and *very crusty*.

For purpose of clarity, we structure our method according to the standard ANFIS architecture (Jang 1993), which proves well suited to implementing the Sugeno model. It is composed of 6 layers (see Fig. 3), which from left to right complete the following tasks: (1) input coding, (2)

computing the related membership degrees to the rules' antecedents, (3) synthesizing them into overall satisfactions of the single rule antecedents, (4) normalizing the above satisfactions on the rule set, (5) computing the effect of the single rules on the consequent according to the Sugeno functions, and (6) combining the single effects into the rule system consequents.

In greater detail, the computations in the various layers are as follows:

- **layer 0**, where we enter the $n = n_c + n_f$ input variables $\mathbf{x} = (\mathbf{y}, \mathbf{g})$, is complicated by the hidden features of those underlying the fuzzy sets (namely, the judgments $\mathbf{g}$). This resolves in the addition of learned shift values δ to initial ones, where the latter may be either the nominal values of the fuzzy numbers used by the tester or completely random numbers (in a reasonable range) computed by the system. The output of this layer is a vector $\tilde{\mathbf{x}}$, where:

$$\tilde{x}_i = \begin{cases} x_i & \text{if } x_i \text{ is a crisp variable } y_i \\ x_i + \delta_i & \text{if } x_i \text{ is a fuzzy variable } g_i \end{cases} \quad (3)$$

Returning to the lead example and using as initial values for the fuzzy sets their nominal values, with reference to the second rule we note that the fuzzy variable $g_1 \equiv \text{very crusty}$ translates into the crisp value obtained by adding to the mode of the fuzzy set A_{22} the learned shift parameter δ_2 . In turn the mode is learned as well by the neuro-fuzzy procedure;

- in the fuzzification **layer 1**, where the membership functions (m.f.s) are introduced, we focus only on triangular functions $\mu_{A_{\{a,b,c\}}}(\tilde{x})$ and asymmetric Gaussian-like functions $\mu_{A_{\{v,\sigma_l,\sigma_r\}}}(\tilde{x})$ which require only three parameters to be specified (Dubois and Prade 1988); namely:

$$\mu_{A_{\{a,b,c\}}}(\tilde{x}) = \begin{cases} \frac{\tilde{x} - a}{(b - a)} & \text{if } \tilde{x} \in [a, b] \\ \frac{c - \tilde{x}}{(c - b)} & \text{if } \tilde{x} \in (b, c] \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_{A_{\{v,\sigma_l,\sigma_r\}}}(\tilde{x}) = \begin{cases} e^{-\frac{(\tilde{x}-v)^2}{\sigma_l^2}} & \text{if } \tilde{x} \leq v \\ e^{-\frac{(\tilde{x}-v)^2}{\sigma_r^2}} & \text{if } \tilde{x} > v. \end{cases} \quad (4)$$

Each membership value reflects how much each input variable matches with the corresponding fuzzy set;

- in **layer 2**, we compute the satisfaction degree w_j of the premise of the j -th rule (the conjunction of the antecedents) as the product of the membership degrees of the metric variables, i.e. using the product T-norm. Remembering that A_{ji} refers to the m.f. fuzzifying the i -th variable in the j -rule, we have:

Fig. 3 The proposed neuro-fuzzy system architecture

$$w_j = \prod_{i=1}^n \mu_{A_{ji}}(\tilde{x}_i) \quad (5)$$

The firing strength of the j -rule reflects how much the variables in input to that rule satisfy its premises;

- in **layer 3**, the normalized satisfaction degree $\bar{w}_j$ is computed as usual by dividing a single degree by the sum of degrees in a same cluster c consisting of k_c rules:

$$\bar{w}_j = \frac{w_j}{\sum_{z=1}^{k_c} w_z} \quad (6)$$

The in-cluster-based normalization is a powerful tool contributing considerably to injecting non-linearities in the overall system, all while preserving the comparable role held by each operational parameter in the task execution;

- in **layer 4**, Sugeno functions f_i are computed as a function of the crisp variables $\tilde{x}$ computed in layer 0 (both y and g , after having learned the shift values modulating the latter). In particular, f_i is composed of a linear combination of suitable (possibly different) scalar functions ζ of each input variable (for instance the identity function, power laws, log, exp, etc.), where the weights s of the linear combination constitute the free parameters to be suitably learned. In the lead example, after having (either randomly or with some expert advice) decided on the shape of scalar functions ζ s, the resulting Sugeno function, which reads $f_2(\tilde{x}, s_1) = s_{21}\tilde{x}_1 + s_{22}\tilde{x}_2^2$ (in case a linear and a quadratic scalar function ζ were chosen for, respectively, $\tilde{x}_1$ and $\tilde{x}_2$), provides the numerical contribution of the second rule to the baking time, reflecting the action to be taken according to the satisfaction degree of its premises. Finally,
- in **layer 5**, the outputs of the Sugeno functions are summed in each cluster c with a weight equal to the normalized satisfaction degree computed at layer 3:

$$o_c = \sum_{j=1}^{k_c} \bar{w}_j f_j(\tilde{x}, s_j) \quad (7)$$

In other words, the more a rule is compatible with the input instance, the higher will be its contribution to the overall output—which reads as the value assigned to the baking time during the next task execution in the lead example.

To sum up, each cluster c in the system, in input the value of both the operational parameters $y^{(t)}$ and the task execution evaluation $g^{(t)}$ (or suitable subsets of them, depending on the operational task to be faced), computes as output the operational parameter $o_c^{(t+1)} \equiv y_c^{(t+1)}$ at the next time step, which in turn receives user evaluation in terms of the fuzzy variables $g^{(t+1)}$. For the sake of conciseness, we will drop the temporal index in the notation whenever no ambiguities arise. Moreover, due to the twofold role played by the operational variables y_c in terms of both input and output of the system, we still refer to the latter with o_c .

We address the entire training task in terms of a back-propagation algorithm (Werbos 1994), hence a long derivative chain as the following:

$$\frac{\partial E}{\partial \theta} = \frac{\partial E}{\partial g} \cdot \frac{\partial g}{\partial o} \cdot \frac{\partial o}{\partial \theta} \quad (8)$$

where θ represents the generic parameter to be learned in the premises and consequents of the FIS, and where scalar symbols were used to lighten the notation. In particular, the error E takes a twofold expression depending on the learning phase (see Table 1). It is the canonical mean square error between original signal τ (the target) and reconstructed signal o in the identification phase, where the former is the value of operational parameters (i.e., baking time in the lead example) which we expect for a given input to the FIS, while the latter is the output proposed by the system. Although in principle the original signal τ may

be suggested by household appliance manufacturers, some remarks on its effective identification in real-world scenarios will be given in the experimental Sect. 3.

On the contrary, in the control phase—which is devoted to the tuning of the FIS according to user needs after suitable identification of the same—error E is assumed to be the square of the task execution evaluation (judgment g).

This choice is motivated by the fact that the set point of g , i.e. the judgment corresponding to complete satisfaction of the user, is 0, while positive/negative deviations from it encompass direction and magnitude of user dissatisfaction. In the lead example, while a consistency of the bread crust satisfying user tastes will be described by the fuzzy judgment “properly crusty”, even though its *true* value remains hidden from both user and system, we assume it to be 0 only in case of full satisfaction (i.e. perfect crust consistency). Each evaluation subtending a consistency crustier than the optimal will be associated to positive shifts from the set point; vice versa in the case of an excessively soft crust. Being the only criteria ruling the *learning from nowhere* inference framework, these order relations introduced on the fuzzy judgments prove to be essential requisites for the success of the inference task.

Note that in the identification phase g is a dummy variable, so that $\frac{\partial E}{\partial g} \cdot \frac{\partial g}{\partial o}$ in (8) contracts in $\frac{\partial E}{\partial o}$, with o , the cluster output, computed as in (7), whereas in the control phase $\frac{\partial E}{\partial g}$ is g itself, modulo a constant. The last derivatives of o w.r.t. all the underlying parameters identifying both membership functions in the premises and regression coefficients in the Sugeno functions are computed as usual [see Masulli et al. (1996); Casalino et al. 1998] for a revisiting of [Nauck et al. (1997), Ch. 8.3]. It is worth noting that in addition to the above parameters, since the *true* values underlying the fuzzy sets responsible for the user evaluation are hidden, we added another parameter, namely the shift δ , to the nominal value of the fuzzy number g [i.e. the central vertex b of the triangular m.f.s or the mean v of the asymmetric Gaussian-like m.f.s in (4)]. While its derivative computation is analogous to the one performed on the nominal value of the fuzzy set, we remind the reader that, contrary to the latter which is

associated to a single membership function, each shift δ refers to each user judgment. So, if the user interacts with the system for m times through a total of n_g evaluation variables, then the total number of δ s parameter will be $m \times n_g$. This fact, which confirms the intrinsic complexity of the learning task w.r.t. standard neuro-fuzzy architectures like ANFIS, highlights how each user judgment is totally decoupled from previous individual evaluations. In this acceptance, the system cannot rely on the existence of presumed relations between two consecutive “very crusty” comments, as this type of linguistic quantifier may refer to two completely different ideas of bread consistency.

As for training modality, we distinguish between either batch mode or cyclic mode (Heskes and Wiegerinck 1996) in the identification phase, thanks to their efficiency and learning stability, and online mode in the control phase, which in turn matches with the real-time mechanism adopted to obtain new instances (operational parameters, user evaluations).

Moreover, according to the *principle of justifiable information granularity* (Pedrycz 2011), which states that a fuzzy set should reflect (or match) the available experimental data to the highest extent, all while being specific enough to come with a well-defined semantic, we further rearranged the inputs $\tilde{x}_i$ s through a proper transformation so as to make them well framed in the corresponding fuzzy sets by filling their support.

Namely, consider the fuzzy sets shown in Fig. 4a where the three triangular m.f.s are related to the linguistic quantifiers “not no crisp”, “properly crisp”, and “very crisp”, while the bullets refer to inputs $\tilde{x}_i$ s processed by the system during a history of user interactions. An initial strategy consists in rescaling the inputs $\tilde{x}_i$ s in one-shot (at each learning cycle) so that their γ and $1 - \gamma$ empirical quantiles would coincide, respectively, with the fuzzy *maxmin* and *minmax* of the three fuzzy numbers (Dubois and Prade 1988). This holds true, respectively, for a proper choice of γ , where *maxmin* gives the maximum of the left extremes delimiting the support of the fuzzy sets, while *minmax* the minimum of the analogous right extremes. In particular, having introduced the affine transformation $\rho(\tilde{x}) = \alpha\tilde{x} + \beta$, at each learning cycle we obtained α by computing $-\frac{q_{1-\gamma}\text{maxmin} - q_\gamma\text{minmax}}{q_\gamma - q_{1-\gamma}}$, and β through $-\frac{\text{minmax} - \text{maxmin}}{q_\gamma - q_{1-\gamma}}$, where q_γ is the γ empirical quantile of the inputs $\tilde{x}$ s. Black bullets in Fig. 4a show the results of the rescaling applied to the gray bullets when γ was set equal to 0.3.

The second strategy, which took the form of an incremental procedure, was the default choice in our experiments, having the obvious advantage to avoid abrupt oscillations in the training process. It is depicted in Fig. 4b in the case of asymmetric Gaussian-like m.f.s. Namely,

Table 1 The analytics of the two-phase learning procedure

Ident. phase	Control phase	
	Analytic	Numeric
$E = (o - \tau)^2$	$E = g^2$	
$\frac{\partial E}{\partial o} = 2(o - \tau)$	$\frac{\partial E}{\partial g} = 2g$	
	$\frac{\partial g}{\partial o} = \left(\frac{\partial o^{(t+1)}}{\partial g^{(t+1)}}\right)^{-1} \approx \frac{g^{(t+1)} - g^{(t)}}{o^{(t+1)} - o^{(t)}}$	
$\frac{\partial o}{\partial \theta} = \text{canonical learning update rule}$		

having introduced for each linguistic variable two translation parameters β', γ' and one scaling parameter α' , we applied the affine transformation:

$$\rho'(\tilde{x}) = \alpha'(\tilde{x} - \beta') + \gamma' \quad (9)$$

to the shifted input value $\tilde{x}$ of each linguistic variable g_i so as to match both position and spread of all the currently learned m.f.s. In particular, we considered a mixture $\mathcal{M}$ of asymmetric Gaussian random variables (John 1982), which differs from the companion set of m.f.s only in the normalization factor, and computed its mean $v_{\mathcal{M}}$ and standard deviation $\sigma_{\mathcal{M}}$ (Robertson and Fryer 1969). At each system evaluation, through a gradient-descent procedure, we gradually modified translation and scaling parameters in (9) to minimize the sum of the mean square errors between the two aforementioned statistics $v_{\mathcal{M}}$ and $\sigma_{\mathcal{M}}$ and their sample realizations computed on the observed data. In Fig. 4b the position of black bullets is the output of several steps of the proposed incremental procedure.

At this point two remarks are noteworthy. Firstly, re the sole user evaluation variables g_s , the output (3) of layer 0 must be further processed so as to include the chosen transformation (ρ or ρ') as the last step. Secondly, we avoided to operate a local rescaling on the single fuzzy sets (e.g. “poorly crisp”) since we would be obliged to make each point $\tilde{x}_i$ belong to a single fuzzy set. While in principle this operation can be straightforwardly performed by attributing a point to the fuzzy set whose membership grade is maximal, this assignment is not free from introducing biases in the learning process. This is due to the continuous adjustment of m.f. parameters and input position during learning.

In conclusion, focusing on asymmetric Gaussian-like m.f.s, our inference concerns the parameter vector θ , with:

$$\theta = (v_{ji}, \sigma_{lji}, \sigma_{rji}, \delta_{pi}, s_{cji}) \quad (10)$$

where v , σ_l and σ_r denote, respectively, mean, left and right standard deviation of the Gaussian m.f. as in (4), δ is the aforementioned shift of each fuzzy input variable, s reads

as the generic regression parameter of the Sugeno function, while c , j and i are indexes coupled, respectively, with the output units, the rules, and the input variables. In the identification phase their increments are ruled as shown in Table 2. Rather, in the control phase the error function reads:

$$E = \sum_i^{n_f} \left(g_i^{(t+1)} \right)^2 \quad (11)$$

where $g_i^{(t+1)}$ is the i -th evaluation provided by the user at time $t + 1$.

Here, the derivative $\frac{\partial g_i^{(t+1)}}{\partial o_i^{(t+1)}}$ is the most critical part of the chain rule (8), since we do not know the mapping linking the FIS output o to the user judgment g . Actually, this mapping is user-dependent, as s/he is the sole actor responsible for this association—consider for example the evaluation as “too crusty” of the currently tasted loaf. Therefore, we propose either:

- an analytical solution in the reciprocal of the derivative $\frac{\partial o_i^{(t+1)}}{\partial g_i^{(t+1)}}$, as clear from the identification of $o_i^{(t+1)}$, or
- a numerical solution in the ratio of the differences at consecutive time steps of the two quantities, i.e. $\frac{g_i^{(t+1)} - g_i^{(t)}}{o_i^{(t+1)} - o_i^{(t)}}$, to bypass identification errors.

The feasibility of the former solution, which in terms of the chain rule (8) reads as:

$$\frac{\partial E}{\partial \theta} = \eta \sum_i^{n_c} \sum_{i'}^{n_f} 2g_{i'}^{(t+1)} \left(\frac{\partial o_i^{(t+1)}}{\partial g_{i'}^{(t+1)}} \right)^{-1} \frac{\partial o_i^{(t+1)}}{\partial \theta} \quad (13)$$

comes directly from the FIS architecture, having in the judgment g at time t one of the inputs of the system, and in the operational parameter o at time $t + 1$ exactly its output. Indeed, referring g to a different time step (exactly the previous one), we see that it suffers from the typical plague of being identified in a domain that may be far from the one where o will be applied during the control phase—a drawback typically overcome by

Fig. 4 **a** Triangular, and **b** asymmetric Gaussian-like fuzzy sets corresponding to the linguistic quantifier “low crisp”, “properly crisp”, and “very crisp”. Gray and black bullets represent the history

of user interactions with the system, respectively, before and after applying the *principle of justifiable information granularity*

Table 2 Back-propagation-like equations ruling the identification phase of the proposed FIS

$$\Delta s_{cji} = \eta(\tau_c - o_c) \bar{w}_j \cdot \varsigma_{cj}(\tilde{x}_i) \quad (12.1)$$

$$\Delta v_{ji} = \begin{cases} \eta \sum_c^{n_c} (\tau_c - o_c) (s_{cji} - o_c) \bar{w}_j (\tilde{x}_i - v_{ji}) / \sigma_{lji}^2 & \text{if } \tilde{x}_i \leq v_{ji} \\ \eta \sum_c^{n_c} (\tau_c - o_c) (s_{cji} - o_c) \bar{w}_j (\tilde{x}_i - v_{ji}) / \sigma_{rji}^2 & \text{if } \tilde{x}_i > v_{ji} \end{cases} \quad (12.2)$$

$$\Delta \sigma_{lji} = \begin{cases} \eta \sum_c^{n_c} (\tau_c - o_c) (s_{cji} - o_c) \bar{w}_j (\tilde{x}_i - v_{ji})^2 / \sigma_{lji}^2 & \text{if } \tilde{x}_i \leq v_{ji} \\ 0 & \text{if } \tilde{x}_i > v_{ji} \end{cases} \quad (12.3)$$

$$\Delta \sigma_{rji} = \begin{cases} 0 & \text{if } \tilde{x}_i \leq v_{ji} \\ \eta \sum_c^{n_c} (\tau_c - o_c) (s_{cji} - o_c) \bar{w}_j (\tilde{x}_i - v_{ji})^2 / \sigma_{rji}^2 & \text{if } \tilde{x}_i > v_{ji} \end{cases} \quad (12.4)$$

$$\Delta \delta_i = \sum_{j=1}^{k_c} \begin{cases} \eta \sum_c^{n_c} (\tau_c - o_c) \left((s_{cji} - o_c) \bar{w}_j \left(-\frac{\tilde{x}_i - v_{ji}}{\sigma_{lji}^2} \right) + \bar{w}_j \frac{\partial s_{cji}}{\partial \tilde{x}} \right) & \text{if } \tilde{x}_i \leq v_{ji} \\ \eta \sum_c^{n_c} (\tau_c - o_c) \left((s_{cji} - o_c) \bar{w}_j \left(-\frac{\tilde{x}_i - v_{ji}}{\sigma_{rji}^2} \right) + \bar{w}_j \frac{\partial s_{cji}}{\partial \tilde{x}} \right) & \text{if } \tilde{x}_i > v_{ji} \end{cases} \quad (12.5)$$

Parameter η denotes the learning rate. The only input variables $\tilde{x}_i$ s coupled with the shifts δ s in (12.5) are the fuzzy judgments g_i s

alternating the two phases (Wu et al. 1992). However, remark to have never been forced to apply this strategy in our experiments.

The second solution

$$\frac{\partial E}{\partial \theta} = \eta \sum_i^{n_c} \left(o_i^{(t+1)} - \tau_i \right)^2 \sum_{i'}^{n_f} 2g_{i'}^{(t+1)} \frac{g_{i'}^{(t+1)} - g_{i'}^{(t)}}{o_i^{(t+1)} - o_i^{(t)}} \frac{\partial o_i^{(t+1)}}{\partial \theta} \quad (14)$$

although temporally consistent, may deserve overfitting, and in any case suffers from the cold start problem.

3 The experimental framework

Granular computing gathers a set of constructs and procedures which have arisen with the goal of matching computational intelligence tools with concrete operational frameworks (Pedrycz and Chen 2011). The interactions of information granules with the physical world and their relation to perception of interactions in the physical world covered by the c -granule paradigm pave the way to concrete solutions of real-world problems. Strictly along these lines, the main experiment testbed of our procedure is framed within the European project SandS, which is aimed at building up a physical and computational networked infrastructure allowing household appliances to better meet the needs of their owners. This is achieved through finely tuned instructions dispatched by a social network to the appliances, in turn equipped with intelligent functionalities. In these settings, the goal is to learn a fuzzy rule system capable of producing proper recipes for our household appliances, where a recipe is the sequence of instructions which completely define the working cycle

of these appliances. As an example, in the context of a loaf taken out of the bread maker, an excerpt of a recipe could be identified by the following ⟨parameter, value⟩ pairs:

⟨first leavening time, 60min⟩

...

⟨baking time, 150min⟩

⟨baking temperature, 190°⟩

...

The goal is to produce recipes that exactly meet needs and preferences of the appliance owner, where the latter are expressed in terms of evaluation on the product/service provided by the appliance, such as loaf moistness and crustiness in the lead example.

In doing this, we have a drawback to eliminate and a challenge to meet.

The drawback concerns the dynamic feature of the entire training procedure. Indeed, since we aim to learn how to get a better user evaluation, we must embed the training phase inside an overall cycle where:

1. a user asks for a task execution;
2. the social network computes the related recipe as a result of previous recipes and related evaluations, and dispatches it to the appliance;
3. the appliance executes the recipe; and
4. the user issues an evaluation on the executed task, so closing the loop (see Fig. 5).

The problem is that step 3 is definitely time consuming (generally, in the order of hours), thus slowing the entire procedure down. This is a pitfall and a severe benchmark as well, since it is met very frequently when we want to

Fig. 5 The online training cycle

apply the soft computing paradigm to real scenarios and not to simulated ones. Hence, the success in managing this problem will decree the success of the entire paradigm; in turn, rendering this procedure successful is the aforementioned challenge that we met with a two-step strategy. On the one hand, we assessed and validated our FIS paradigm on a case study—namely, the face beautifying task—which has the same features yet almost null recipe execution time (see Sect. 3.1). On the other hand, after a brief explanation of the SandS project and architecture, we apply in Sect. 3.2 the procedure to the actual task of producing recipes for our appliances.

3.1 The case study: face beautification

The face beautifying task is described in the following algorithm.

- Start from a picture of a slightly deformed face like in Fig. 6a;
- **do**
 1. a tester is requested to express a $\{-5, \dots, +5\}$ Likert evaluation (Trochim 2006) of the picture w.r.t. the four criteria listed in Table 3b;
 2. on the basis of these evaluations, the FIS algorithm (replaced by a human operator in the first 120 runs) may change the anthropomorphic parameters reported in Table 3a;
 3. a graphical routine updates the face picture according to these changes, obtaining variations like in Fig. 6b;
 4. the new picture is supplied to the tester for a new evaluation, so that the procedure continues from step 1;
- **until** either a satisfactory picture is achieved (not necessarily close to the original one, see Fig. 6c), or a given stopping condition is reached.

Interested readers may play with the online tool², meanwhile contributing to the enrichment of the continually growing benchmark. At the same time, they may have first-hand experience of the intrinsic complexity of the task. In fact, despite the immediacy and at first glance apparent local scope of the anthropomorphic parameters, actually they have been planned so as to be responsible for a global morphing of the face. They are also highly correlated with one another, so that a slight modification of one parameter has an effect on face elements in principle ruled by any other parameters. Analogously, the employed evaluation criteria are not simply a matter of subjective interpretations. In addition, their relation with anthropomorphic parameters is highly non-linear and non-univocal, in the sense that various parameter settings may collapse in the same evaluation, even when performed by the same user.

Thanks to the online tool, at a rate of 20 iterations per hour, we may rely on circa 1000 triplets (task, recipe, evaluation) in approximately one week, where:

- tasks may be differentiated on the basis of both the involved tester and the face to be enhanced: we may in fact suppose that the evaluation answers differ both from tester to tester on a same picture, and from face to face on the same tester;
- recipes will be generated through slight modifications of the current anthropomorphic parameters, thanks to fuzzy rules properly tuned by the learning algorithm on the basis of the tester fuzzified evaluations. An excerpt of a prototypical recipe could be:

$\langle \text{facewidth}, +3.26 \rangle$
 $\langle \text{faceheight}, -2.41 \rangle$
 $\dots$
 $\langle \text{noselength}, +1.63 \rangle$

² http://mockup.laren.di.unimi.it/test_facedeform/.

Fig. 6 Morphing of a face **Table 3** List of anthropomorphic parameters and evaluation criteria employed in the face beautification task (Van Dongen 2014)

(a)		(b)	
par ₁	face width	eval ₁	approachability
par ₂	face height	eval ₂	youthfulness
par ₃	face roundness	eval ₃	dominance
par ₄	mouth thickness	eval ₄	golden ratio
par ₅	mouth width		
par ₆	mouth height		
par ₇	eye size		
par ₈	distance between		
par ₉	eyes slant of eyebrows		
par ₁₀	nose width		
par ₁₁	nose length		

where we used as value of each pair the shift the corresponding anthropomorphic parameter should be affected by;

- users provide their evaluations at each image modification through a suitable Likert scale.

In this way we may plan to gather operational indications for the training of FIS on our actual appliances' testbed. Namely, in both cases the evaluation criteria, expressed in the discrete range $\{-5, \dots, +5\}$, guide the learning system in the selection of a dozen *operational* parameters, each affecting the results of the process (pleasant face in one case, good bread, soft laundry, etc. in the other one). Analogously, since the relation underlying user evaluation and operational parameters is unknown, we will rely on a set of rules, synthesizing a general wisdom, whose premises are mainly fuzzy while their consequences are metric. Also for the face experiment we will use the proposed special Sugeno system where some of the metric variables in input to the premises (precisely those referred to the user evaluations) are hidden.

3.2 The goal testbed

In an extreme synthesis, we are setting up the SandS ecosystem (Apolloni et al. 2013, 2015) which deals with a social network aimed at producing recipes with tools of computational intelligence, to be dispatched to household

appliances grouped in homes via a domestic WiFi network. A recipe is a set of scheduled, possibly conditional, instructions (hence a sequence of evaluated parameters such as water temperature or soak duration) which completely define the running of an appliance. They are managed by a home middleware to be properly transmitted to the appliance via suitable protocols.

The entire contrivance is devised to optimally carry out usual housekeeping tasks through a proper running of home appliances with a minimal user intervention. Feedbacks are sent by the housekeepers and appliances themselves to a networked intelligence module (in the role of a virtually electronic super-mom in Fig. 7) to close the continuous recipe optimization loop. In this cadre, experts and appliance manufacturers may contribute with off-line advices and suggestions. An electronic board interfaces each single appliance to the home middleware.³ An excerpt of a recipe for washing machine is reported in Table 4. It consists of two tables; with the former we fix the parameters that characterize the working cycle to be executed by the machine; with the latter we list the sequence of instructions that are sent step by step to the machine to implement this working cycle.

A relevant aspect concerns the network intelligence. In our approach, starting from the most suitable baseline recipe, we modify some of its parameters on the basis of the feedback it received on its execution. We do so through a rule set that has been trained on the log of the triples $\langle \text{task, recipe, evaluation} \rangle$. In this way we translate our inference problem on the recipe parameter variations into the one of identifying the rules generating the recipes. In addition, since the underlying rules are based in great part on vague evaluations expressed by the user both on the past recipe performance and on the current task specification, also in this case we fall in the field of fuzzy rule systems with *hidden* variables, proper to the *learning from nowhere* framework. Actually, analogously to Table 3, in Table 5 we list the operational parameters and evaluation criteria in the bread-making task.

³ See <https://www.youtube.com/watch?v=8NHzmXR1Jck> for a snapshot tutorial.

Fig. 7 The SandS super-mom **Table 4** An excerpt of washing machine recipe

(a)				(b)	
				Instructions	
Phases	Parameters			DOOR, LOCK	
Pre Washing	Temperature	Water	Time	LABEL, PREWASHING	
Soaking	Water	Time		WATER, BOX3	
Washing	Temperature	Water	Time	LEVEL, 10	
Rinsing	Water	Water	Cycles	ON_LEVEL, 600	
Spinning	Spin	Time		HEATER, ON	
				TEMPERATURE, 15	
				ON_TEMP, 120	
				ALTERNATE, 7	
				SPIN, 150	
				WAIT, 120	
				MOTOR, OFF	
				...	

(a) The operational parameters that characterize each phase

(b) The sequence of instructions setting their value in the machine board

4 Numerical results

4.1 The face beautification experiment

At present, we have performed two kinds of experiments by computing either a subset or the entire set of face parameters. Moreover, since the procedure we propose is relatively new, we focused on the same face beautifying task to conduct many numerical investigations on the constituent steps of the algorithm, with special focus both on the positioning of the hidden variables under the fuzzy sets and on the identification of the Sugeno functions and their inverses too.

4.1.1 Turing like test

As for the former, like in the Turing Test (Turing 1950), in our experiment we let 6 parameters—mainly related to nose and eyes shaping—be tuned by the user, and the other 5—regulating the overall face dimensions and

mouth—be produced by our fuzzy rule system, with the aim that the two families of parameters would result indistinguishable to an external observer. First of all, we obtained a suitable training set by letting users interact with the online face beautification tool.⁴ Namely, each user drove the beautification process by cyclically adjusting the 11 anthropomorphic parameters and then evaluating the pleasantness of the resulting face. In this way, we got a collection of 121 records, each deputed to describe each beautification task through a list of (face parameters, evaluation) histories. On the basis of the above training set, the identification phase proved satisfactory, as shown by the Mean Square Error (MSE) curves reported in Fig. 8.

Accordingly, Fig. 9 shows the correspondence between the original parameters set by the user in the 121 records of the training set and the companion ones computed by the fuzzy rule systems. The 45 degree

⁴ <http://mockup.laren.di.unimi.it/facedeform/>.

Table 5 List of operational parameters and evaluation criteria employed in the bread-making task (Mondal and Datta 2008)

(a)		(b)	
par ₁	first leavening time	eval ₁	fragrance
par ₂	first leavening temperature	eval ₂	softness
par ₃	second leavening time	eval ₃	baking
par ₄	second leavening temperature	eval ₄	crustiness
par ₅	pre-baking time		
par ₆	pre-baking temperature		
par ₇	baking time		
par ₈	baking temperature		
par ₉	Turning golden time		
par ₁₀	Turning golden temperature		

straight lines highlight the good approximation of the recovered points.

As for the FIS architecture used throughout this experimentation, we peaked randomly the components c_i s of each Sugeno function f among linear and quadratic functions, committing the FIS for a proper identification of the related coefficients. We used the asymmetric Gaussian-like m.f.s in (4) and affine transformation (9) to let the coordinates of the linguistic variables fulfill the support of the Gaussian bells. Clusters of rules were used to singularly generate shifts the current values of anthropomorphic parameters are affected by. Each cluster was composed of 3^5 rules, where 3 is the number of levels of the adopted fuzzy quantifiers (e.g., with reference to the approachability evaluation, they read as “low”, “somewhat”, and “very approachable”, standing, respectively, for “aloof”, “apathetic”, and “friendly”), and 5 is the number of antecedents in the rules: the 4 evaluations plus the questioned parameter computed in the previous run. Namely, for purposes of homogeneity with the other antecedents, we decided to group into three fuzzy sets the values assumed by the single questioned parameter in the training set (using the same three-level quantifiers as the ones adopted for evaluation variables), and to specialize the rules for each fuzzy quantifier and for each cluster as well.

The control phase consists of a recall of the identified system, still subject to a continuous training based on the sole user evaluations. Namely, having as target the judgment the user makes on the pleasant appearance of the current face, and modifying its parameters on the basis of the history of user interaction in terms of pairs (face parameters, evaluations), the fuzzy system is called upon to generate new face parameters, with the aim of moving user judgments around their set points. Prior to a more rigorous evaluation procedure, we let a dozen users evaluate the pleasantness of the face generated by the FIS after about 15 iterations. They all said that they were highly satisfied. Fig. 10 reports the trajectories of parameters and

evaluations along a session where three different faces were submitted to one of these users. In all instances both parameters and evaluations converged to the optimal value (set equal to 0), after some wandering in the respective spaces. In fact, observing a full user satisfaction in correspondence to no modifications, in output to the FIS, of the values of the operational parameters, means that a suitable equilibrium point has been achieved by the system.

A different trend is denoted by the parameters when their setting is left in the hands of the user. In Fig. 11 we observe less spread in their values but no convergence to the neutral settings at the end of the three training sessions outlined in the picture.

4.1.2 The complete task

The second experiment, involving the entire set of parameters, has similar operational features but weaker identification performances, as denoted, for instance, by the two recovering graphs in Fig. 12.

However, the user is able to drive the image toward the face in Fig. 13a, which he assumed to be satisfactory, through the parameters' paths in Fig. 13b. The fact is that, during the face beautifying session, a two-way adaptation process occurs: FIS adapts to the user evaluations and user adapts to the FIS behavior. This process is shorter (around 15 steps) when the system is well identified in the batch phase, and longer (23 steps) when starting from a less accurate FIS. It is exactly this reciprocal adaptation process that we rely on in the transfer from the case study to the appliances' regulation instances.

For a preliminary comparative study, we considered a simple Actor-Critic Reinforcement learning procedure (Grondman et al. 2012) as a competitor of our method. In a true essence, the Actor is the user replaced by the system which decides how to improve a face. The Critic is definitely the user who evaluates the effects of the parameter changes chosen to improve the face. Per se, the

Fig. 8 Course of the Mean Square Error (MSE) during FIS identification phase

Fig. 9 Comparison between the original parameters set by the user in the 121 records of the training set (x axis), and the companion ones computed by the fuzzy rule systems (y axis)

procedure results in a random walk in the parameter space with two sagacities:

1. it gives very small perturbations to the already trained system;

2. it drives the chance in a very conservative way, relying on a backtracking procedure as the last option.

The pseudocode of the entire procedure is reported in Algorithm 1.

Algorithm 1 Reinforcement Learning face beautification pseudocode.

- 1: define the default variation range $[e_{j0}^{\text{inf}}, e_{j0}^{\text{sup}}]$, symmetric around 0, for each parameter x_j (for both signal y_j and judgment g_j). Set the current range $[e_j^{\text{inf}}, e_j^{\text{sup}}] = [e_{j0}^{\text{inf}}, e_{j0}^{\text{sup}}]$.
- 2: **for** each parameter y_j **do** draw a random variation δ_j in the range $[e_j^{\text{inf}}, e_j^{\text{sup}}]$
- 3: evaluate the outcome recipe execution
- 4: **if** the appreciation is positive **then** expand the variation extremes having the same sign of the related δ by a quantity proportional to δ
- 5: **if** the appreciation is negative **then** reduce the variation extremes having the same sign of the related δ by a quantity proportional to δ
- 6: **if** $e_j^{\text{sup}} - e_j^{\text{inf}}$ is greater than a given threshold ϑ_1 for almost all j **then goto** 2
 else set the variation extremes to the default values (*long jump*: $[e_j^{\text{inf}}, e_j^{\text{sup}}] = [e_{j0}^{\text{inf}}, e_{j0}^{\text{sup}}]$)
- 7: **for** each parameter x_j **do** draw a random variation δ_j in the range $[e_j^{\text{inf}}, e_j^{\text{sup}}]$
- 8: **if** the outcome evaluation is positive **then goto** 2,
 else backtrack and draw new variations
- 9: **if** the negative evaluation is greater than a given threshold ϑ_2 **then goto** 2
- 10: **if** the number of negative evaluations is below a given threshold ϑ_3 **then goto** 2
 else stop

The lead strategy is to encourage the exploration of the parameter space each time this produces a good result and to compress the exploration in the opposite case. When the exploration gets stuck in a local optimum, so that explorations are more and more inhibited (parameter variation ranges close to 0), we call for a *long jump* by resetting the variation ranges to their default values. If the jump is inappropriate (negative evaluation) then a backtracking procedure is performed, and another jump is realized. Finally, the algorithm stops after too many negative jumps.

The face beautification experiment shows improvements in a trend passing from phenotypes to genotypes. The Critic bases its actions on the user's likes/dislikes (the phenotype). However, it may root its computations on the ranges of the parameter changes on the used parameters (the genotypes). The core of the above procedure lies in the criterion: if changes give rise to improvements (hence further likes) then maintain a large range, exactly in the directions where the modifications have been affected. Hence, if a parameter has been decremented with success

(a new user like) decrease the negative extreme of the effected changes. Rather, if it gave rise to a dislike, then increase the negative extreme. Analogously with the positive extreme. The local minimum trap may be discovered simply by thresholding the change ranges: when the range of each parameter goes below a given threshold (hence, after many dislikes), we reset the ranges to the initial values, thus favoring a long jump in another parameters' region. If no jump is effective, then stop.

Fig. 11 Parameters trajectories observed during the beautification process carried out without the FIS support

Fig. 10 Parameters and evaluations trajectories observed during the beautification process carried out with the FIS support

Fig. 12 Same graphs as in Fig. 9, when all 11 parameters are jointly identified

Fig. 13 Face beautification when all 11 parameters are jointly identified: **a** the face obtained after 23 evaluations performed by the user; **b** the companion parameters computed by the fuzzy rule system

We pay for the simplicity of the algorithm with a longer set of trials before getting satisfactory results. Fig. 14 shows this aspect as a companion of Fig. 13. We may see the squeezed thread of the parameters due to the reset of their range after the sticking in a local minimum.

4.2 Adequacy, rationality and generality of the proposed procedure

We used the same dataset collected for the face beautifying task to check the reasonableness and correctness of the various steps constituting our learning algorithm. In particular, we focus here on the adequacy of the procedure in discovering the hidden variables, which in turn represents the main novelty of the proposed FIS. Namely, with reference to the same control phase described in detail in the previous sections, for each judgments gs we were asked to find some regularities with the corresponding hidden variables hs learned by the system. To this aim, we randomly drew initial values for the latter in the range $[-5, 5]$ and coupled each h with a judgment g , ranging in the integer lattice $\{-5, \dots, +5\}$. Representing each judgment g through three fuzzy quantifiers (namely, “low”, “medium” and “high”), we run the learning algorithm by adjusting both hidden variables hs and parameters of the triangular m.f.s so as to achieve full user satisfaction. Fig. 15 denotes the good performance of the procedure in discovering the hidden variables, highlighting the preservation of the monotonicity between the values expressed

by the users (y axis), namely the judgments gs , and the level of the fuzzy set identified within the three m.f.s by the one denoting the maximum membership of h (x axis). In view of the initial random coupling between hidden variables hs and user judgments gs , this is a remarkable result reflecting the general requirement of monotonicity on the maps considered in (Pedrycz et al. 1997). These experiments and emerging properties indicate that our method is very general and robust, so as to be employable in many operational fields for purposes of eliminating our blindness re the universe of discourse. Indeed, we simply transfer the above procedures to the goal testbed without noteworthy changes.

4.3 Facing the original task

Moving on to our recipe generation task, we applied our two-phase procedure to the bread maker experiment dataset.⁵ It was generated by letting the partners of the SandS project prepare the loaf with the bread machine and then asking a suitable set of tasters to evaluate the quality of the home-baked bread. Repeating this experiment each day over a period of 4 months, we obtained a total of 200 pairs (operational parameters, evaluations). As clear from Table 5, in this case we have 10 parameters (time and temperature in the 5 baking phases) and 4 evaluations as

⁵ The database is available at <http://ns3366758.ip-37-187-78.eu/exp/excel>.

Fig. 14 Same pictures as in Fig. 13 when the parameter exploration is drawn by Reinforcement Learning

Fig. 15 Monotonicity preservation in the hidden variables

well. Clusters of rules were used to singularly generate the operational parameters. The overall FIS architecture was the same as the one adopted for the face beautification task (refer to Sect. 4.1 for more details).

As our first training instance, we omitted training the FIS on parameters 6, due to its scarce variability, and 10, since temperature settings above the security threshold (set to 150 in the microcontroller) are dummy. Rather, the identification on the remaining parameters (after proper normalization) proved sufficiently satisfactory (see Fig. 16).

However, many records in the training set have been collected during a sort of users' training phase, so that their suggestions about the new recipe—which constitutes the FIS target on the identification phase—may definitely prove misleading. Hence we decided to dope the training set with a set of examples made up of the original instances as to the premises and of the optimal parameter settings (which we learned during the experimental campaign) as to the consequences. This expedient was profitable, as confirmed by the mean square error curves reported in Fig. 17.

Essentially the system is pushed toward the optimal parameters in the consequents, profiting from a well-biased noise represented by the original examples. With this FIS configuration the system produces a good tuning of the bread maker parameters even on new inputs (hence in

generalization). However, to render the system really adaptive to the judgments of the single user so as to learn how to get a better evaluation from her/him, we needed to train the FIS following the single user interactions depicted in Fig. 5, as we did with the face beautification case study. Here the problem is a little trickier due to the slowdown incurred by the entire process, which in turn is caused by the delays introduced by the appliance during the recipe execution. Hence we devised the following expedient. After a good system identification (performed as before), we may assume that FIS computes the correct parameters in the doped part of the training set, so that the evaluation given in correspondence of these parameters properly applies to the computed parameters as well. So, we may use this part of the training set to train the system on the evaluations as well (the control phase). Rather, we adopted an intermediate strategy by pairing the identification error with the control error through a mutual weighting based on a linearly convex composition. Namely, with reference to Table 1, the error becomes:

$$E = \xi \sum_i^{n_f} g_i^2 + (1 - \xi) \sum_i^{n_c} (\tau_i - o_i)^2 \quad (15)$$

for proper (small enough) $\xi \in [0, 1]$. Now that the FIS has been weaned, it is ready to work online. In this case, the

Fig. 16 Plots of the tested operational parameter values and their recovering in the bread-making experiment

error expression remains the same, where the second term acts as a regularization term, since the target τ now is the previous parameter value and ξ is more or less big, depending on the evaluation values. The criterion is: if the

overall evaluation is close to 0, then refrain FIS from producing new parameters that are far from the current ones. Vice versa do not hesitate to move far from them if the current evaluation is poor (nowhere near 0). Table 6

Fig. 17 Mean square error descent on the 10 operational parameters of the doped training set

reports a log of these online interactions leading to a gentle reduction of the leavening and baking parameters in response to user evaluations, expressed in terms of slight defects of softness, baking, and crustiness.

5 Conclusions

In this paper, we solve a new learning instance, that we denote by learning from complex granules, in response to a new characterization of information granules in terms of fuzzy sets on a hidden universe of discourse and their relations with the physical environment. What emerges is a new Granular Construct that by means of cognitive tools solves the problem of learning the parameters of a single c -granule. It is framed between the two constructs investigated so far: one where crisp variables are fuzzified to enter

a fuzzy rule, and the other where fuzzy sets are dealt with as a whole within a fuzzy rule. The learning procedure is an extension of the back-propagation algorithm running on an architecture that is rather complex as for both the questioned variables and the two (identification/control) operational phases. Robustness of the numerical results and weakness of the theoretical guarantees are inherited by the original algorithm. Currently, we claim no comparative efficiency w.r.t. other approaches to the problem, apart from a trivial comparison with Reinforcement Learning in the case study. Rather, we stress its suitability as a very general method for concretely solving problems that are frequently met in the real life. In the sector of white goods manufacturers this is witnessed by the conduction of household appliances in a true fuzzy framework, where the principal inputs come from user reactions. In a tight concreteness thread, we considered all operational aspects of the specific

Table 6 Log of online user interactions with the system

1st Leaven.		2nd Leaven.		Pre-Baking		Baking		Turning golden		Evaluation			
Time	Temp.	Time	Temp.	Time	Temp.	Time	Temp.	Time	Temp.	Fragr.	Softn.	Baking	Crust
1821	33	2788	35	575	59	3885	118	101	126	+3	+1	+1	+1
1792	33	2813	35	569	59	3872	118	100	126	0	+2	+2	+2
1820	33	2787	35	570	59	3885	118	97	126	0	+1	+1	+1
1766	33	2754	34	560	59	3889	117	93	124	0	+2	+2	+2

problem of training a bread maker, and found the proof of its solution in a record of people daily preparing their loaves of bread according to the suggestions of the trained system.

Future work will be devoted to exploring in greater depth the theoretical aspects of the paradigm we have introduced. The main questions to answer are:

1. To what extent may we reduce the data inside the information granule? Obviously, this sentence may be read in many ways, questioning what we denote by datum, what by information granule. In a very pragmatic approach, we may ask ourselves about the sense of ruling a bread maker on the basis of feedback from a kook. In this case we may have a virtual reality where the user is satisfied, along with an effective reality where the same user may be intoxicated by the bread his machine has baked.
2. To what extent we may make our *c*-granule a more complex granule? We must recall that a distinguishing feature of our granule is that the *soft_suit* is rooted in a hidden universe of discourse, a drawback that we try to overcome thanks to its strong relationship with the signals coming from the *hard_suit* and an efficient *link_suit*. Parameterizing the granules in terms of approximation spaces, for instance, requires measures on the universe of discourse that would constitute an additional target of our inference system. The additional complexity opens the path to interesting abstract operations on granules, such as composition, hierarchy, etc., which become crucial in field such as Big Data Pedrycz and Chen (2015). As we mentioned in Sect. 2, for now we privileged a pure subsymbolic way based on the cognitive adjustment of non-symbolic parameters. However, the exploration of a hybrid approach will be a relevant part of our future work.

References

- Amershi S, Cakmak M, Knox WB, Kulesza T (2014) Power to the people: the role of humans in interactive machine learning. *AI Magazine* 35(4). doi:10.1609/aimag.v35i4.2513
- Apolloni B, Fiaschè M, Galliani GL et al (2013) Social things—the SandS instantiation. In: Second IEEE Workshop on the Internet of Things: Smart Objects and Services, IoT-SoS 2013
- Apolloni B, Galliani GL, Gioia M et al (2015) Social things: now we can. *Intell Artif* 9:25–41
- Apolloni B, Pedrycz W, Bassis S, Malchiodi D (2008) The puzzle of granular computing, studies in computational intelligence, vol. 138, 1 edn. Springer, Berlin, Heidelberg
- Casalino F, Masulli F, Sperduti A (1998) Rule specialization in networks of fuzzy basis functions. *Intell Autom Soft Comput* 4(1):73–82
- Clark PG, Grzymala-Busse JW, Rzas W (2013) Generalizations of approximations. In: Lingras P, Wolski M et al (eds) *Rough sets and knowledge technology. Proceedings of 8th International Conference RSKT 2013*, Halifax, NS, Canada, October 11–14, 2013. LNCS, vol. 8171. Springer, pp 41–52
- Cococcioni M, Lazzerini B, Marcelloni F (2007) Estimating the concentration of optically active constituents of sea water by takagisugeno models with quadratic rule consequents. *Pattern Recognit* 40(10):2846–2860
- Dubois D, Prade H (1988) Fuzzy numbers, an overview. In: Bezdek JC (ed) *Analysis of fuzzy information (Mathematics)*. CRC Press, Boca Raton, FL, pp 3–39
- Fuller R (1988) Fuzzy reasoning and fuzzy optimization. No. 9, TUCS General Publications, Turku, Centre for Computer Science, Abo
- Grondman I, Busoniu L, Lopes G, Babuska R (2012) A survey of actor-critic reinforcement learning: standard and natural policy gradients. *IEEE Trans Syst Man Cybern Part C Appl Rev* 42(6):1291–1307
- Heller M (1990) The ontology of physical objects: four-dimensional hunks of matter. Cambridge studies in philosophy. Cambridge University Press, Cambridge
- Heskes T, Wiegierinck W (1996) A theoretical comparison of batch-mode, on-line, cyclic, and almost-cyclic learning. *IEEE Trans Neural Netw* 7(4):919–925
- Hofmann T, Schölkopf B, Smola AJ (2008) Kernel methods in machine learning. *Ann Stat* 36(3):1171–1220
- Jang JSR (1993) ANFIS: adaptive-network-based fuzzy inference system. *IEEE Trans Syst Man Cybern* 23(3):665–685
- Jankowski A, Skowron A (2013) Practical issues of complex systems engineering: Wisdom Technology Approach. Springer, Heidelberg
- Jankowski A, Skowron A, Swiniarski R (2013) Interactive rough-granular computing in wisdom technology. In: Yoshida T, Kou G et al (eds) *Active media technology, LNCS, vol 8210*. Springer International Publishing, Switzerland, pp 1–13
- John S (1982) The three-parameter two-piece normal family of distributions and its fitting. *Commun Stat Theory Methods* 11(8):879–885
- Jordan MI, Rumelhart DE (1992) Forward models: supervised learning with a distal teacher. *Cogn Sci* 16(3):307–354
- Kacprzyk J, Pedrycz W (eds) (2015) *Handbook of computational intelligence*. Springer-Verlag, Berlin, Heidelberg
- Mamdani EH (1977) Application of fuzzy logic to approximate reasoning using linguistic synthesis. *IEEE Trans Comput* 26(12):1182–1191
- Masulli F, Sperduti A, Alfonso D (1996) Hybrid pattern-recognition scheme. In: Bruno Bosacchi JCB (ed) *Proceedings of SPIE 2761, Applications of Fuzzy Logic Technology III*. Orlando, Florida, pp 154–162
- Mendel JM, John RIB (2002) Type-2 fuzzy sets made simple. *IEEE Trans Fuzzy Syst* 10(2):117–127
- Miu S, Hayashi Y (2000) Neuro-fuzzy rule generation: survey in soft computing framework. *IEEE Trans Neural Netw* 11:748–768
- Mondal A, Datta AK (2008) Bread baking—a review. *J Food Eng* 86(4):465–474
- Nauck D, Klawonn F, Kruse R (1997) *Foundations of neuro-fuzzy systems*. John Wiley & Sons Inc, New York
- Pedrycz W (2011) The principle of justifiable granularity and an optimization of information granularity allocation as fundamentals of granular computing. *J Inf Process Syst* 7(3):397–412
- Pedrycz W (2013) *Granular computing analysis and design of intelligent systems*. CRC Press, Boca Raton, FL
- Pedrycz W, Chen SM (2011) *Granular computing and intelligent systems: design with information granules of higher order and higher type*. Springer, Berlin, Heidelberg
- Pedrycz W, Gudwin R, Gomide F (1997) Nonlinear context adaptation in the calibration of fuzzy sets. *Fuzzy Sets Syst* 88(1):91–97

- Pedrycz W, Chen SM (eds) (2015) Granular computing and decision-making, interactive and iterative approaches. Springer International Publishing
- Provotar AI, Lapko AV, Provotar AA (2013) Fuzzy inference systems and their applications. *Cybern Syst Anal* 49(4):517–525
- Rabiner L, Juang B (1986) An introduction to hidden Markov models. *IEEE Acoustics Speech Signal Process Mag* 3:4–16
- Robertson CA, Fryer JG (1969) Some descriptive properties of normal mixtures. *Scand Actuar J* 1969(3–4):137–146
- Skowron A, Jankowski A (2015) Interactive granular computing. In: Ciucci D, Wang G et al (eds) *Rough sets and knowledge technology, LNCS*, vol 9436. Springer International Publishing, Switzerland, pp 50–61
- Skowron A, Stepaniuk J, Swiniarski R (2012) Modeling rough granular computing based on approximation spaces. *Inf Sci* 184(1):20–43
- Skowron A, Szczuka M (2010) Toward interactive computations: a rough-granular approach. In: Koronacki J, Ra Z et al (eds) *Advances in machine learning II, studies in computational intelligence*, vol 263. Springer, Berlin Heidelberg, pp 23–42
- Takagi T, Sugeno M (1985) Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans Syst Man Cybern SMC* 15(1):116–132
- Trochim WM (2016) Likert scaling. *Research Methods Knowledge Base*, 2nd edn. <http://www.socialresearchmethods.net/kb/concepts.php>
- Turing AM (1950) Computing machinery and intelligence. *j-MIND* 59(236):433–460
- Van Dongen S (2014) Associations among facial masculinity, physical strength, fluctuating asymmetry and attractiveness in young men and women. *Ann Hum Biol* 41(3):205–213
- Werbos PJ (1994) The roots of backpropagation. From ordered derivatives to neural networks and political forecasting. John Wiley & Sons, Inc., New York
- Wu Q, Hogg B, Irwin G (1992) A neural network regulator for turbogenerators. *IEEE Trans Neural Netw* 3(1):95–100
- Zadeh LA (1979) A theory of approximate reasoning. In: Hayes J, Michie D, Mikulich LI (eds) *Machine intelligence*, vol 9. Halstead Press, New York, pp 149–194
- Zadeh L (1965) Fuzzy sets. *Inf Control* 8:338–353