Photons detection in Positron Emission Tomography through Iterative Rule Learning of TSK rules

Ismael Rodríguez-Fdez, Manuel Mucientes, Alberto Bugarín
Dep. Electronics and Computer Science
University of Santiago de Compostela
{ismael.rodriguez, manuel.mucientes, alberto.bugarin.diz} @usc.es

Abstract— **A correct reconstruction of medical images in Positron Emission Tomography (PET) needs a precise estimation of the position of the incidence photons in the detector surface. The traditional method based on Anger algorithm calculates the position as a polynomial of the intensities. However, it fails to track the true position near the edges of the detector. In this paper Takagi-Sugeno-Kang (TSK) fuzzy rules are used in order to obtain gradual smooth outputs for different situations with a consequent represented in a similar way as Anger does. The algorithm that learns the TSK rules is based on the Iterative Rule Learning approach. The learned knowledge bases have been tested with a set of Monte Carlos simulations of PET photon detection. Results show a good performance of our proposal, which has been compared with other approaches.**

Keywords— **Iterative Rule Learning; Genetic Algorithm; Takagi-Sugeno-Kang; Fuzzy Rules; Positron Emission Tomography;**

I. Introduction

A correct reconstruction of medical images in Positron Emission Tomography (PET) needs a precise estimation of the position of the incidence photons in the detector surface. Usually, the reconstruction of these images is very sensitive to inevitable measurement errors. These errors could be reduced increasing the resolution of the detectors. However, this also increases the cost of the PET device. Therefore, it is essential to have an estimation system as reliable as possible, able to improve the detection performance with the same hardware.

The most popular technique to predict the position of photons on a PET detector is known as Anger method. This method consists in a linear combination of the intensity distribution. The anger method is quite accurate at the center of the detector, but fails near the edges. Different algorithms have been proposed to improve the position determination [4]: squaring the calculation of Anger method, based on the skewness of the light distribution and by means of an iterative optimization algorithm for the solution of a regularized nonlinear least squares problem. Also, neural networks based on multi-layer perceptrons [10] and radial basis function network [1] have been used in order to improve the estimation error.

Our proposal takes into account two premises: high accuracy, but providing the expert with valuable and easy to extract information. Fuzzy rules have the capability to obtain a smooth result interpolating the output of each fired rule in a gradual way. Moreover, Takagi-Sugeno-Kang (TSK) fuzzy rules [13], [12] have an adequate structure for the requirements of the problem. In a TSK rule, the output is obtained as a polynomial of the measured the input variables. So, if the input variables verify the antecedent part of the rule, the light distribution belongs to that class with a certain degree, and the position of the photon can be estimated with the polynomial in the consequent of the rule, in a similar way as Anger does.

Learning of fuzzy knowledge bases through the use of evolutionary algorithms has shown to be a powerful technique [2]. Evolutionary algorithms have two main advantages over other learning methods in this field: first, the rules of the knowledge base can be coded in many different ways, due to the flexibility in the representation of the solutions. On the other hand, another important advantage is that we can manage the tradeoff between interpretability and accuracy of the learned rules through the use of different algorithms.

The main contributions of the paper are: (i) the proposed algorithm is able to learn TSK fuzzy rules with high precision for regression problems improving the Iterative Rule Learning (IRL) approach; (ii) the proposal is able to learn rules involving linguistic labels with multiple granularity for regression; (iii) the results on PET data shows a great performance over other existing methods.

The paper is structured as follows: Sec. II explains in more detail the photons detection in PET, while Sec. III describes the algorithm that has been used to learn the TSK rules. Sec. IV presents the results and, finally, Sec. V points out the conclusions.

II. Detection of photons in PET

PET is a non-invasive diagnostic method of medical imaging which allows displaying metabolic activity in a slice of the body by means of detecting radiation, emitted from a radio-isotope injected into the body of the patient. A high-energy photon emitted

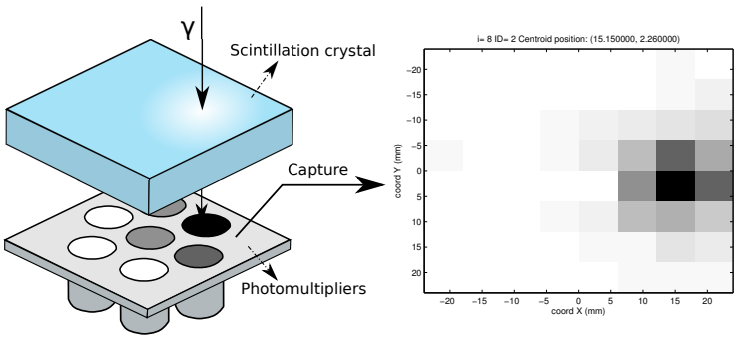

Fig. 1. Detection of a photon.

from the body collides with the scintillation crystal, in which it produces a shower of low-energy visible photons. These photons are collected by a set of photomultipliers coupled to the crystal, allowing to reconstruct the incidence point from the evoked potentials. Then, this information is converted into the slice image by a reconstruction algorithm.

Detection of the high-energy photons in PET is a crucial stage prior to image reconstruction. The ability to precisely calculate the photon coordinates from electric pulses invoked in the photomultipliers implies high detection resolution and allows better reconstruction accuracy.

The most common method for scintillation point estimation is the Anger algorithm. In this method, the scintillation point coordinates are computed according to:

$$\begin{aligned} x &= \sum_{i,j=1}^{N} Q(i,j) \cdot x_i \\ y &= \sum_{i,j=1}^{N} Q(i,j) \cdot y_i \end{aligned} \tag{1}$$

where Q is the matrix of the light distribution whose elements $Q(i,j)$ represent the number of detected optical photons, $(x_i, y_i), i, j = 1, \ldots, N$ are the coordinates of $Q(i,j)$, and $N \times N$ is the number of pixels within the detector. This method works well at the center of the detector, where the light distribution is entirely known. However, it fails to track the true position near the edges of the detector.

Fig. 1 shows an example of detection of a photon in (15, 3) point. The image of the light intensities shows a truncation to the right of the photon : only one column of intensities is used in order to displace the coordinate estimation to the right against 6 columns that displace the coordinate estimation to the left. Therefore, the estimation method defined by Anger have wrong results in this situations.

III. Learning Algorithm

A. Evolutionary Learning of Knowledge Bases

According to [3], [6], evolutionary learning of knowledge bases has different approaches to represent the solution to the problem: Pittsburgh, Michigan, IRL [2], and GCCL.

In the IRL approach, the whole rule base is learned by the evolutionary algorithm. After each sequence of iterations, the best rule is selected and added to the final rule base. The selected rule must be penalized in order to induce niche formation in the search space. A common way to penalize the obtained rules is to delete the training examples that have been covered by the set of rules in the final rule base. The final step of the IRL approach is to check whether the obtained set of rules is a solution to the problem. In the case it is not, the process is repeated. A weak point of this approach is that the cooperation among rules is not taken into account when a rule is evaluated.

Our proposal is based on IRL. The learning process is divided in epochs (set of iterations), and at the end of each epoch a new rule is obtained. The following sections describe each of the stages of the algorithm.

1: $KB_{cur} := \varnothing$
2: **repeat**
3: $it := 0$
4: $equal_{ind} := 0$
5: Initialization
6: Evaluation
7: **repeat**
8: Selection
9: Crossover and Mutation
10: Evaluation
11: Replacement
12: **if** $best^{i-1}_{individual} = best^{i}_{individual}$ **then**
13: $equal_{ind} := equal_{ind} + 1$
14: **else**
15: $equal_{ind} := 0$
16: **end if**
17: $it := it + 1$
18: **until** $(it < it_{min} \vee equal_{ind} < it_{check}) \wedge (it < it_{max})$
19: $KB_{cur} := KB_{cur} \cup best_{ind}$
20: $uncov_{ex} := uncov_{ex} - cov^{best}_{ex}$
21: **until** $uncov_{ex} = \varnothing$
22: Rule Subset Selection

Fig. 2. IRL algorithm.

B. Examples

The learning process is based on a set of training examples obtained from photons detection. Each detection generates a matrix Q of dimension $N \times N$ that represents the distribution of light intensities. Moreover, the simulation process gives the actual photon coordinates. Using these data, the following steps are performed in order to generate the examples for the evolutionary algorithm:

1. For each detection, two examples are generated: one using the data from the columns in order to estimate the X coordinate and the other using the data from the rows for the Y coordinate.
2. First, the accumulated intensity for each column (row) is calculated.
3. Then, the accumulated intensities are normalized so that their sum is equal to 1 and the column (row) of the higher intensity is labeled the reference ref.
4. The number of columns (rows) that are picked for each example is $n < N$.

(a) The number of column (row) intensities selected to the right of ref is defined as:

$$right_n = \begin{cases} \frac{n}{2} & if\ \frac{n}{2} + ref \leq N \\ N - ref & otherwise \end{cases} \quad (2)$$

(b) On the other hand, the number of column (row) intensities selected to the left of ref is defined as:

$$left_n = n - right_n \quad (3)$$

5. Finally, the output coordinate is the deviation of the photon detection coordinate from the center point of ref.

So, in this paper, each example e^l is represented by a tuple:

$$e^l = (q_1,\ \ldots,\ q_n,\ o) \quad (4)$$

where q_i is the normalized intensity for the column (row) i, n is the number of i-th columns (rows) and o is the relative output.

C. Individual Representation

Each individual of the evolutionary algorithm represents a TSK fuzzy rule. Therefore, each individual is divided into two parts:

- The antecedent, which consists in n linguistic labels representing the n intensities of the examples (F_q).
- The consequent, which is formed by the weights p of the polynomial that represents the output of the TSK rule:

$$p_1 q_1 + p_2 q_2 + \ldots + p_n q_n + p_0 \quad (5)$$

The linguistic labels of the antecedent (F_q) are defined using a multiple granularity approach. The universe of discourse of a variable is divided into a different number of labels for each granularity. Specifically, a granularity g^i_{var} divides the variable var in i uniformly spaced labels, i.e., $A^i_{var} = \{A^{i,1}_{var}, ..., A^{i,i}_{var}\}$. Fig. 3 shows a partitioning up to granularity five. On the other hand, the p weights of the consequent are defined as real numbers.

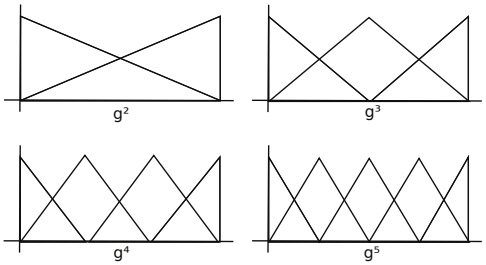

Fig. 3. Multiple granularity approach.

D. Evaluation

The fitness of an individual is calculated as follows. First, the examples that are not covered by the final knowledge base (line 19, fig. 2) are defined as:

$$uncov_{ex} = \{e^l\ :\ DOF_{KB_{cur}}(e^l) < DOF_{min}\} \quad (6)$$

i.e. the set of examples that are covered with a degree of fulfillment below DOF_{min} by the current final knowledge base. Then, $uncov_{ex}$ is divided into two groups: the set of examples that are covered by individual j, defined as:

$$cov^j_{ex} = \{e^l_u : DOF_{R_j}(e^l_u) > DOF_{min}\}, \quad (7)$$

where $DOF_{R_j}(e^l_u)$ is the degree of fulfillment of e^l_u for rule j, e^l_u is an example that fulfills $e^l \in uncov_{ex}$; and the set of examples that are not covered by individual j:

$$uncov^j_{ex} = \{e^l_u : DOF_{R_j}(e^l_u) < DOF_{min}\}, \quad (8)$$

The fitness of an individual in the population is calculated as the combination of two elements. On one hand, the accuracy with which the individual covers the examples, called confidence. On the other hand, the ability of generalization of the rule, called support. The confidence can be defined as:

$$confidence = \frac{1}{1 + \frac{\sum_l error^l_j \cdot DOF_{R_j}(e^l)}{\sum_l DOF_{R_j}(e^l_u)}} \quad (9)$$

where $error^l_j$ is calculated as:

$$error^l_j = (o^l - C^l_{R_j})^2 \quad (10)$$

where o^l is the value of the output of example e^l and $C^l_{R_j}$ is the output of the rule codified in the individual j. Note that for confidence, the $error^l_j$ is calculated for all the examples in the training set, giving a measure of the global error for all the examples covered by the individual j.

Support is calculated as:

$$support = \begin{cases} 1 & if\ \frac{\#cov^j_{ex}}{\#uncov_{ex}} \geq cov_{ref} \\ \exp\left(\frac{\#cov^j_{ex}}{\#uncov_{ex}} - cov_{ref}\right) & otherwise \end{cases} \quad (11)$$

where $\frac{\#cov_{ex}^{j}}{\#uncov_{ex}}$ is the percentage of examples in $uncov_{ex}$ that are covered by the individual j and cov_{ref} is a parameter that indicates the percentage of reference.

Finally, we define $fitness$ as a linear combination of both values:

$$fitness = \alpha_f \cdot confidence + (1 - \alpha_f) \cdot support \quad (12)$$

which represents the strength of an individual over the uncovered examples. $\alpha_f \in [0,\ 1]$ is a parameter that codifies the trade-off between accuracy and generalization of the learning algorithm.

E. Initialization

An individual is generated for each example in the training set. A hill-climbing approach is used in order to initialize the antecedent. On the other hand, for the consequent part a Ridge regression [5] is performed in order to obtain the weights. Each individual is generated using the following steps:

1. The F_q linguistic labels are initialized with granularity g_q^1 and $confidence_{best}$
2. n neighbor antecedents are generated, incrementing the granularity of one of the linguistic labels (a different one for each neighbor). An step is performed for each antecedent incrementing the granularity. This is, n different antecedents are generated: $(ant_1, \ldots, ant_n)$
   (a) For each new antecedent ant_i, select the examples e^l that fulfill $DOF_{ant_i}(e^l) >= DOF_{min}$
   (b) These examples are used for the Ridge regression in order to obtain the weights of the consequent.
   (c) The $confidence$ is calculated using the Ridge weights.
3. The individual with best $confidence$ is selected as $confidence_{sel}$.
4. If $confidence_{sel} > confidence_{best}$ then go to step 2, else finish initialization and return the best individual.

F. Crossover

The matching of the pairs of individuals that are going to be crossed is implemented following a probability distribution defined as:

$$P_{close}(\alpha, \beta) = \frac{1}{1 + \frac{C_{sim}}{M}} \quad (13)$$

where C_{sim} represents the similarity of the outputs for the examples covered for α or β:

$$C_{sim}(\alpha, \beta) = \sum_l |C_\alpha^l - C_\beta^l| :\ DOF_{R_\alpha}(e_u^l) \geq DOF_{min} \ or\ DOF_{R_\beta}(e_u^l) \geq DOF_{min} \quad (14)$$

and M is the number of examples that fulfill $DOF_{R_\alpha}(e_u^l) \geq DOF_{min}$ *or* $DOF_{R_\beta}(e_u^l) \geq DOF_{min}$

With this probability distribution, the algorithm selects with higher probability mates that have similar consequents. The objective is to extract information on which propositions of the antecedent parts of the rules are important, and which are not. The crossover operator generates two offspring:

$$offspring_1 = crossover(ind_\alpha, ind_\beta)$$
$$offspring_2 = crossover(ind_\alpha, ind_\beta) \quad (15)$$

Crossover performs two different operations: antecedent crossover, and consequent crossover. For the antecedent crossover, the following steps are performed:

1. First, select randomly an antecedent F_q^{sel}.
2. Then, an operation must be picked depending on the existence of that antecedent in both individuals (table I):

- If the antecedent is not taken into account (i.e., the granularity of F_q^{sel} is g_q^1) in the first individual but it exists (granularity differs from g_q^1) in the second one, then the proposition of the second individual is copied to the first one, as this proposition could be meaningful.
- If the situation is the opposite to the previous one, then the proposition of the first individual is deleted, as it might be not important.
- If the antecedent exists in both individuals, then both propositions are combined in order to obtain a proposition that generalizes both antecedents.

TABLE I
CROSSOVER OPERATIONS

Ind. 1	Ind. 2	Action
no	yes	add antecedent in individual 1
yes	no	delete antecedent
yes	yes	combine antecedents

In this last case, the combination of antecedents is done taking into account the degree of similarity between them (Fig. 4), where similarity [11] is calculated as:

$$similarity(F_\phi, F_\psi) = 1 - \frac{\sum_x |\mu_\phi(x) - \mu_\psi(x)|}{card(X)} \quad (16)$$

where F_ϕ and F_ψ are the labels to compare and X is a set of finite points x over $support(\phi \cup \psi)$.

Only when both similarities are partial, the antecedents are merged:

- If the similarity is 0, then the antecedents correspond to different situations. This means that the proposition of the first individual might not contain meaningful information and it could be deleted to generalize the rule.
- If the similarity is total, then, in order to obtain a new individual with different antecedents, the proposition is eliminated.
- Finally, if the similarity is partial, then the propositions are merged in order to obtain a new one that combines the information provided by the two original propositions. Thus, the individual is generalized. The merge action is defined as the process of finding the label with the highest possible granularity that

has some similarity with the labels of both original propositions.

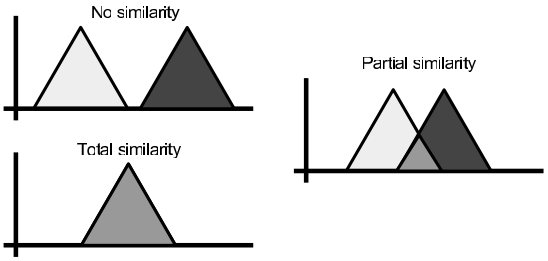

Fig. 4. Different possibilities of similarity in the crossover

After the crossover of the antecedents, the consequent is also changed. Ridge regression is performed over the covered examples (this set has changed due to mutation), and PCBLX is executed using Ridge and the previous consequent.

When the antecedent crossover is not performed, the consequents are selected to be crossed. The crossover operator for the consequents is the parent-centric BLX (PCBLX) [7], [9]. Given two real-coded chromosomes, $X = (x_1 \dots x_g)$ and $Y = (y_1 \dots y_g)$ $(x_i, y_i \in [a_i, b_i], i = 1, \dots, g)$, that are going to be crossed, the following offspring are generated (Fig. 5):

- $Z = (z_1 \dots z_g)$, where z_i is randomly selected from the interval $[l_i^z, u_i^z]$, with $l_i^z = max\{a, x_i - I\}$, $u_i^z = min\{b_i, x_i + I\}$, and $I_i = |x_i - y_i| \cdot pcblx_\alpha$, $pcblx_\alpha \in [0, 1]$.
- $V = (v_1 \dots v_g)$, where v_i is randomly selected from the interval $[l_i^v, u_i^v]$, with $l_i^v = max\{a, y_i - I\}$, $u_i^v = min\{b_i, y_i + I\}$, and $I_i = |x_i - y_i| \cdot pcblx_\alpha$, $pcblx_\alpha \in [0, 1]$

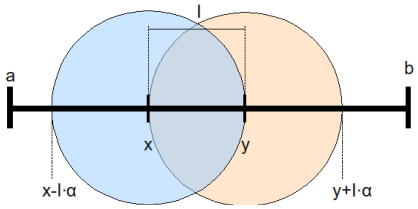

Fig. 5. PCBLX operator for real numbers.

G. Mutation

When crossover is not performed, both individuals are mutated. Mutation implements two different strategies: generalize or specialize a rule. The higher the value of the confidence (Eq. 9), the higher the probability to generalize the rule by mutation. This occurs with rules that cover their examples with high quality and that could be modified to cover other examples. On the contrary, when the confidence of the individual is low, this means that it is covering some of its examples with a low performance. In order to improve the rule, some of the examples that are currently covered should be discarded to get a more specific rule.

For generalization, the following steps are performed:

1. Select an example $e^{sel} \in uncov_{ex}^j$ (Eq. 8). The example is selected with a probability distribution given by the error calculated as:

$$P(C_{R_j}|e_u^l) = exp(-error_j^l) \tag{17}$$

The higher the similarity between the output of the example and the consequent of rule j, the higher the probability to be selected.

2. The individual is modified in order to cover e^{sel}. Therefore, all the propositions with $\mu_{prop}\left(e^{sel}\right) < DOF_{min}$ are selected for mutation. These are the propositions that are not covering the example.

3. The selected propositions are generalized, choosing the most similar label in the adjacent partition with lower granularity. The process is repeated until $\mu_{prop}\left(e^{sel}\right) > DOF_{min}$.

For specialization, the process is equivalent:

1. Select an example $e^{sel} \in cov_{ex}^j$ (Eq. 7). The example is selected with a probability distribution that is inversely proportional to $P\left(C_{R_j}|e_u^l\right)$ (Eq. 17). The higher the similarity between the output of the example and the consequent of rule j, the lower the probability of being selected.

2. Only one proposition needs to be modified to specialize, and it is selected randomly.

3. For the selected proposition, specialization is done by choosing the most similar label in the adjacent partition with higher granularity until $\mu_{prop}\left(e^{sel}\right) < DOF_{min}$.

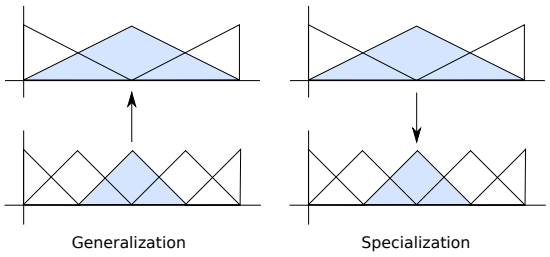

Fig. 6. Generalization/Specialization examples

After the mutation of the antecedents the consequent is also mutated. Ridge regression is performed over the covered examples (this set has changed due to mutation), and PCBLX is executed using Ridge and the previous consequent.

H. Selection and replacement

Selection has been implemented with a binary tournament selection. On the other hand, replacement follows an steady-state approach. The new individuals and those of the previous population are joined, and the best $pop_{maxSize}$ individuals are selected for the next population.

I. Epoch loop

An epoch is a set of iterations at the end of which a new rule is added to KB_{cur}. The stopping criterion of each epoch (inner loop in fig. 2) is the

number of iterations, but this limit varies according to the following criteria: once the number of iterations reaches it_{min}, the algorithm stops if there are it_{check} consecutive iterations with no change in $best_{ind}$. If the number of iterations reaches it_{max} iterations, then the algorithm stops regardless of the previous condition.

When the epoch ends, rule $best_{ind}$ is added to KB_{cur}. Moreover, the examples that are covered by $best_{ind}$ (according with Eq. 7) are marked as covered by the algorithm (line 20, fig. 2). Finally, the algorithm stops when either no uncovered examples remain or no example is covered by the individual learned in the last epoch.

IV. Results

The proposed algorithm has been tested using 8,599 examples generated by Monte Carlo simulation of photons detection. The examples have been divided into different sets, corresponding to each of the combinations of $right_n$ and $left_n$, as shown in table II. For each one, a fuzzy knowledge base was learned. When an example is not covered by the corresponding knowledge base, Anger method is performed instead. Table II shows the size of the examples for each combination.

TABLE II
Size of the datasets

$left_n$	$right_n$	#examples
3	3	4623
4	2	4225
5	1	3986
6	0	4359

Figure 7 shows a rule learned by the proposed algorithm for $right_n = left_n = 3$. An example that fits with the situation indicated by the antecedent part is showed in figure 8. In order to obtain the input data, the steps of section III-B are followed. The *Acc* column indicates the accumulated intensity for each row, while $Norm$ shows the normalized values so that their sum is equal to 1. In this case, the fifth row is labeled as the reference (ref) and the normalized intensities of the three preceding and following rows are used as the input data.

The consequent of the rule is the deviation of the photon detection coordinate from the center point of ref (2.5). To calculate it, the rows close to ref (q_3 and q_4) have the largest weights, while for the farthest ones the weight is close to zero. Moreover, the sign of the weights indicates to which side the photon detection is deviated from the center point. Positive values suggest that the photon detection coordinate is below the reference, while negative values indicate that it is over the reference. For this example, the rule is fired with a degree of 0.204 and the output deviation is 2.811, resulting in a value of the y coordinate of the photon detection equal to 5.811 $(2.5 + 2.811)$.

IF
q_1 is $A_{q_1}^{1,1}$ and q_2 is $A_{q_2}^{2,1}$ and q_3 is $A_{q_3}^{4,1}$
and q_4 is $A_{q_4}^{3,2}$ and q_5 is $A_{q_5}^{4,1}$ and q_6 is $A_{q_6}^{1,1}$
THEN
$o = 0 \cdot q_1 - 4.908 \cdot q_2 - 10.427 \cdot q_3$
$+9.241 \cdot q_4 + 7.587 \cdot q_5 + 0 \cdot q_6 - 0.003$

Fig. 7. A typical rule learned by the algorithm.

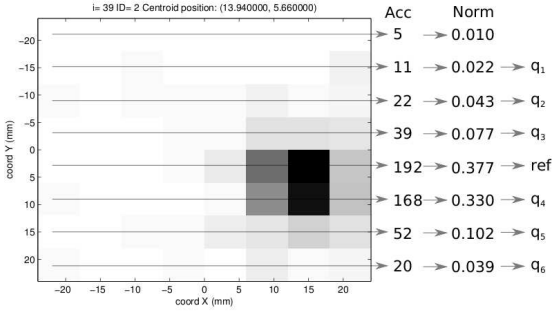

Fig. 8. An example that fires the in rule Fig. 7

The results shown in table III have been obtained as an average of a cross-validation approach with 5 folds. For each of each of the combinations of $right_n$ and $left_n$, three values are represented: $\bar{\chi}$, σ_{cv} and $\bar{\sigma}_{run}$. $\bar{\chi}$ is the arithmetic mean over 25 executions (five-fold cross-validation with 5 runs). σ_{cv} is the standard deviation over the arithmetic means of each data partition, and represents the robustness of the algorithm to obtain similar results regardless the data partition. Finally $\bar{\sigma}_{run}$ is the arithmetic mean of the standard deviations over the 5 runs for all the data partitions, which reflects the robustness of the probabilistic algorithm to obtain similar results regardless the followed pseudo-random sequence.

The values that have been used for the parameters of the evolutionary algorithm are: $n = 6$, $pcblx_\alpha = 0.1$, $DOF_{min} = 0.0001$, $\alpha_f = 0.9$, $pop_{maxSize} = 100$, $it_{min} = 50$, $it_{check} = 10$, $it_{max} = 100$, $prob_{cross} = 0.8$ and $prob_{ant} = 0.75$ (probability to cross the antecedent part instead of the consequent one).

The first column in table III indicates the dataset. The last four columns show the results obtained by the following algorithms:

- IRL: the proposal described in this paper.
- IRL+ILS: the proposal described in this paper with a post-processing stage for the selection of rules to improve the collaboration between them based on the Iterated Local Search (ILS) algorithm[8].
- Anger: the traditional method for estimation of coordinates in PET.
- NN: a Multi-layer Perceptron neural network with six neurons in the hidden layer.

Moreover, table IV shows the number of rules obtained by our proposal for the different datasets. The best results on test for each of the datasets have been highlighted in bold face (table III). The Anger

TABLE III
RESULTS OF THE DIFFERENT ALGORITHMS FOR SEVERAL DATASETS

			IRL	IRL+ILS	Anger	NN
3-3	Training	X	0.808	0.801	0.882	0.708
		cv	0.095	0.099	-	0.130
		run	0.095	0.099	-	0.131
	Test	X	**0.824**	0.825	-	1.156
		cv8	0.392	0.391	-	0.705
		run	0.392	0.392	-	0.903
4-2	Training	X	0.406	0.394	0.984	0.336
		cv	0.031	0.030	-	0.016
		run	0.032	0.030	-	0.020
	Test	X	0.428	**0.417**	-	0.690
		cv	0.140	0.140	-	0.518
		run	0.140	0.140	-	0.566
5-1	Training	X	0.563	0.529	3.734	0.466
		cv	0.063	0.050	-	0.031
		run	0.064	0.051	-	0.032
	Test	X	0.588	**0.542**	-	0.900
		cv	0.165	0.187	-	0.549
		run	0.166	0.187	-	0.670
6-0	Training	X	1.578	1.545	16.184	0.561
		cv	0.046	0.051	-	0.033
		run	0.052	0.059	-	0.033
	Test	X	1.672	1.626	-	**0.649**
		cv	0.211	0.226	-	0.188
		run	0.214	0.230	-	0.204

method shows a good performance on datasets 3-3 and 4-2. However, as the truncation degree increases, its performance drastically diminishes. This is especially true for dataset 6-0, where the error is of more than 16mm on average.

The best results in three of the four datasets are obtained by our proposal (IRL+ILS), although in the first dataset IRL without selection obtains an slightly better performance. The neural network is only competitive for the most complex dataset (6-0), where it obtains the best result. However, in the other datasets the performance is not adequate. In summary, our proposal shows a good performance and consistent results over the different datasets.

TABLE IV
SIZE OF THE LEARNED RULE BASES

		IRL	IRL+ILS
3-3	X	32.3	26.2
	cv	10.1	11.4
	run	18.9	19.6
4-2	X	8.2	2.4
	cv	1.3	0.9
	run	2.3	1.6
5-1	X	13.8	7.3
	cv	1.8	2.6
	run	3.5	3.6
6-0	X	18.2	14.6
	cv	4.8	3.4
	run	5.1	4.6

V. Conclusion

An algorithm, based on an improved version of the IRL approach to learn fuzzy rules, has been presented. The algorithm has been applied to learn a knowledge base for the estimation of the photon position emission in PET. Our proposal uses TSK fuzzy rules for regression, and the learning process involves linguistic labels with multiple granularity. The algorithm has been tested with several simulation photon captures showing a good performance in comparison with the standard technique in the field of PET.

Acknowledgment

This work was supported by the Spanish Ministry of Science and Innovation under grants TIN2008-00040, TIN2011-22935 and TIN2011-29827-C02-02. I. Rodriguez-Fdez is supported by the Spanish Ministry of Science and Innovation, under the FPU national plan. M. Mucientes is supported by the Ramón y Cajal program of the Spanish Ministry of Science and Innovation.

References

[1] A.M. Bronstein, M.M. Bronstein, M. Zibulevsky, and Y.Y. Zeevi. Optimal nonlinear line-of-flight estimation in positron emission tomography. *Nuclear Science, IEEE Transactions on*, 50(3):421–426, 2003.

[2] O. Cordón and F. Herrera. Hybridizing genetic algorithms with sharing scheme and evolution strategies for designing approximate fuzzy rule-based systems. *Fuzzy sets and systems*, 118:235–255, 2001.

[3] O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena. *Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases*, volume 19 of *Advances in Fuzzy Systems - Applications and Theory*. World Scientific, 2001.

[4] A. Cornelio, F. Gasperini, S.L. Meo, N. Lanconelli, S. Moehrs, S. Marcatili, MG Bisogni, and A.D. Guerra. Comparison of different reconstruction methods for planar images in small gamma cameras. *Journal of Instrumentation*, 6:C01030, 2011.

[5] T. Hastie, R. Tibshirani, and J.H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Verlag, 2009.

[6] F. Herrera. Genetic fuzzy systems: Taxonomy, current research trends and prospects. *Evolutionary Intelligence*, 1:27–46, 2008.

[7] F. Herrera, M. Lozano, and A.M. Sánchez. A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. *International Journal of Intelligent Systems*, 18:309–338, 2003.

[8] H.R. Lourenço, O.C. Martin, and T. Stützle. *Handbook of Metaheuristics*, chapter Iterated Local Search, pages 321–353. Kluwer Academic Publishers, 2003.

[9] M. Lozano, F. Herrera, N. Krasnogor, and D. Molina. Real-coded memetic algorithms with crossover hill-climbing. *Evolutionary Computation*, 12:273–302, 2004.

[10] F. Mateo, RJ Aliaga, N. Ferrando, JD Martínez, V. Herrero, C.W. Lerche, RJ Colom, JM Monzó, A. Sebastiá, and R. Gadea. High-precision position estimation in pet using artificial neural networks. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 604(1-2):366–369, 2009.

[11] B. Vantaggi R. Scozzafava. Fuzzy inclusion and similarity through coherent conditional probability. *Fuzzy Sets and Systems*, 160:292–305, 2009.

[12] M. Sugeno and G.T. Kang. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28:15–33, 1988.

[13] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15:116–132, 1985.