HAL
open science

Highly Interpretable Linguistic Knowledge Bases Optimization: Genetic Tuning versus Solis-Wetts. Looking for a good interpretability-accuracy trade-off

J.M. Alonso, O. Cordon, Serge Guillaume, L. Magdalena

To cite this version:

J.M. Alonso, O. Cordon, Serge Guillaume, L. Magdalena. Highly Interpretable Linguistic Knowledge Bases Optimization: Genetic Tuning versus Solis-Wetts. Looking for a good interpretability-accuracy trade-off. IEEE Conference Proceedings, 2007, 11278, p. 901 - p. 906. hal-00453834

HAL Id: hal-00453834

https://hal.science/hal-00453834v1

Submitted on 5 Feb 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

"This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder." IEEE Conference Proceedings - ISSN: 1098-7584 http://www.ieee.org/web/publications/rights/policies.html

Highly Interpretable Linguistic Knowledge Bases Optimization: Genetic Tuning versus Solis-Wetts. Looking for a good interpretability-accuracy trade-off

José M. Alonso, O. Cordón, S. Guillaume, and L. Magdalena

***Abstract*— This work shows how to achieve a good interpretability-accuracy trade-off through keeping the strong fuzzy partition property along the whole fuzzy modeling process. First, a small compact knowledge base is built. It is highly interpretable and reasonably accurate. Second, an optimization procedure, which only affects the fuzzy partitions defining the system variables, is carried out. It improves the system accuracy while preserving the system interpretability. Two optimization strategies are compared: Solis-Wetts, a local search based strategy; and Genetic Tuning, a global search based strategy. Results obtained in a well-known benchmark medical classification problem, related to breast cancer diagnosis, show that our methodology is able to achieve knowledge bases with high interpretability and accuracy comparable to that obtained by other methodologies.**

I. Introduction

Fuzzy Logic [25] is acknowledged by its well known flair for linguistic concept modeling. The semantic expressivity of Fuzzy Logic (FL), using linguistic variables [26] and linguistic rules [18], is quite close to expert natural language. As a result, the use of FL favours the interpretability of the final model, but does not guarantee it. For that reason, there are works with the aim of setting restrictions to the fuzzy modeling process in order to guarantee the interpretability of the fuzzy model finally obtained. For example, [23] establishes semantic constraints for membership functions. On the other hand, other proposals [7] are dedicated to improve the interpretability of fuzzy systems.

This paper focuses on classification problems where interpretability is of prime concern, such as diagnosis problems. Accuracy, at least at a given level, is a prerequisite. To be worthy of consideration, the system has to be accurate enough. On the contrary the rules wouldn't be considered as pieces of knowledge. Anyhow, priority is also given to interpretability. In some cases, both criteria can be satisfied to a high degree, but in most cases it is not possible. They are conflicting goals; high accuracy usually means low interpretability and *vice versa*. Finding a good trade-off between accuracy and interpretability is one of the most difficult tasks in system modeling. Two main trends are found in the fuzzy modeling literature regarding that trade-off. On the one hand, those who are first focused on the interpretability and then try to improve the accuracy [6]. On the other hand, those who build a knowledge base (KB) focused on the accuracy and then try to improve its interpretability [7]. According to the classification made in [2], the first approach is called Linguistic Fuzzy Modeling with improved accuracy, and the second one is known as Precise Fuzzy Modeling with improved interpretability.

Of course, systems built from expert knowledge, where a domain expert is able to describe the system behavior, are highly interpretable. Moreover, the expert knowledge is usually a general knowledge related to the most influential variables and the global system behavior. Alternatively, systems can also be built using experimental data, which are likely to give a good image of interaction between variables. However, the induced knowledge from data is always a specific knowledge related to the situations described in the available data set. Both kinds of knowledge convey complementary information, and their cooperation is likely to yield compact systems with a high performance.

Thanks to the fuzzy logic formalism, induced knowledge can be described with the same kind of linguistic variables and rules than those used for expressing expert knowledge. A new methodology for combining both kinds of knowledge was proposed in [16]. Its implementation is called HILK (**H**ighly **I**nterpretable **L**inguistic **K**nowledge bases) and it includes integration, simplification, consistency analysis, optimization, and evaluation processes. The present paper is focused on the optimization phase. Two different optimization strategies are analyzed and compared to tune the fuzzy system membership functions. Starting from a compact KB with high interpretability and an acceptable accuracy, the goal is to improve the interpretability-accuracy trade-off, increasing accuracy but preserving interpretability without altering the strong fuzzy partition property.

The structure of the paper is as follows. Section II describes the methodology proposed for building highly interpretable knowledge bases. Section III shows two different tuning methods, Solis-Wetts and Genetic Tuning. Section IV explains the experiments made and the obtained results. A well-known benchmark classification problem, Wisconsin breast cancer, has been tackled with the aim of comparing the two optimization approaches. Finally, section V offers some conclusions.

José M. Alonso is with the Technical University of Madrid, Ciudad Universitaria s/n, 28040 Madrid, Spain (email: jmam@mat.upm.es).

O. Cordón and L. Magdalena are with the European Centre for Soft Computing, Edificio Científico-Tecnológico, C/. Gonzalo Gutiérrez Quirós s/n, 33600 Mieres, Asturias, Spain (email: oscar.cordon@softcomputing.es, luis.magdalena@softcomputing.es).

S. Guillaume is with the Cemagref Montpellier, BP 5095, 34196 Montpellier Cedex 5, France (email: serge.guillaume@montpellier.cemagref.fr).

II. Highly Interpretable Linguistic KBs

The three conditions for a fuzzy rule-based system (FRBS) to be interpretable have been stated in [15]:

1) Use of linguistic variables with interpretable fuzzy partitions. Each system variable is described by a set of linguistic terms, modeled as fuzzy sets. The use of strong fuzzy partitions [21] satisfies semantic constraints [23] (distinguishability, normalization, coverage, overlapping, etc.) on membership functions. Figure 1 shows a strong fuzzy partition (SFP) with 5 terms. The granularity for each variable should be kept small enough to make the system accurate while being understandable. According to psychologists, 7 ± 2 is a limit on human information processing capability [19]. A SFP satisfies the next conditions:

$$\forall x \in U,\ \sum_{i=1}^{M} \mu_{A_i}(x) = 1 \tag{1}$$

$$\forall A_i\ \exists x,\ \mu_{A_i}(x) = 1 \tag{2}$$

where U=[U_l, U_u] is the universe of discourse, U_l and U_u are the lower and upper limits respectively, M is the number of linguistic terms and $\mu_{A_i}(x)$ is the membership degree of x to the A_i fuzzy set.

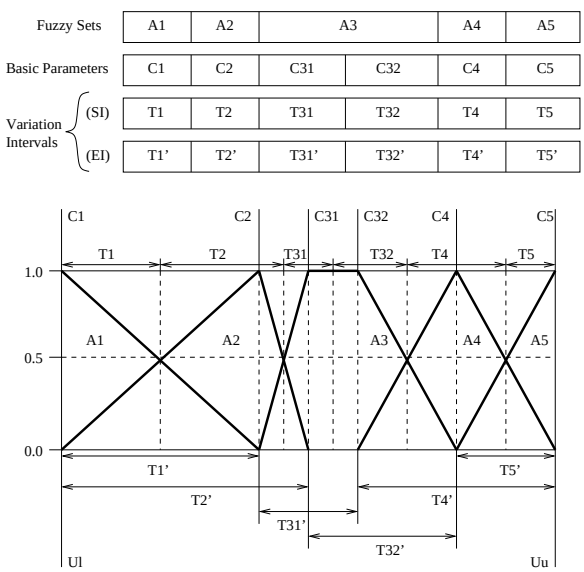

Fig. 1. A strong fuzzy partition

2) Use of a small number of linguistic rules. The system behavior is described by means of linguistic rules in the form **If** *condition* **Then** *conclusion* where both, the condition and conclusion use linguistic terms. The condition part is made up of tuples (*input variable*, *linguistic term*), where the absence of an input variable in a rule means that the variable is not considered in the evaluation of the rule.
3) Use of compact rules for large systems. A fuzzy rule is compact if its premise is defined by a subset of the input variables.

HILK methodology lets us build highly interpretable linguistic KBs. The cooperation framework was proposed in [16], and its implementation consists of the next steps:

- Defining the most influential variables according to expert knowledge and experimental data. The expert can provide complete or partial information about the variables (*Expert Partitions*). On the other hand, fuzzy partitions can be created from data (*Induced Partitions*). SFPs are kept along the whole process.
- Building a common universe for each variable, according to both expert knowledge and data distribution. The integration of all available knowledge for partition design is made previous to the rule definition.
- Describing the system behavior through linguistic rules. The expert is invited to express his/her system knowledge as linguistic rules (*Expert Rules*). Also, rules are built from data (*Induced Rules*).
- Integrating both expert and induced rules into the rule base. Thanks to the common universe previously defined, both types of rules use the same linguistic terms defined by the same fuzzy sets. As a consequence, rule comparison can be done at the linguistic level. During this last step, rule integration, the fundamental properties of a rule base have to be guaranteed.

The expert is supposed to be able to assess induced knowledge. Hence, the whole integration process at both levels, partitions and rules, is run under his/her control. Three main steps are carried out regarding the rule base integration:

- First, a consistency analysis of the rule base, and the subsequent process for solving the linguistic conflicts previously detected.
- Second, a simplification procedure which increases interpretability keeping either consistency or accuracy.
- Third, an optimization process with the aim of increasing accuracy, but maintaining interpretability.

As the first two steps are thoroughly explained in [3] and [16] this work focuses on the last one. Let us now go into details about it.

III. Optimization process

The optimization phase only affects the fuzzy partitions that define the system variables. It comes to membership function tuning. It is constrained in order to maintain the SFP property. Two strategies were studied:

1) An element by element optimization procedure based on the classical local search strategy proposed by Solis and Wetts [22]: It is *a hill climbing method with memorization of the previous successes* [13]. The goal is not to find the global optimum, but to improve accuracy by performing a few iterations. Two cases are analyzed: Variable by variable, and label by label.
2) An *all-in-one* optimization procedure based on a global search strategy inspired on the evolutionary processes that take place in nature, a genetic algorithm (GA) [14]. In our case, it becomes a genetic tuning process [10]. GAs usually start with a population of several randomly generated solutions, chromosomes, and get better solutions by applying genetic operators. All system parameters are adjusted at the same time.

In both cases, the coding scheme considered is the same. The partition basic parameters (fuzzy sets centers or modal points, C_i points in figure 1) are adjusted through slight modifications to increase the system accuracy, but preserving meaningful fuzzy sets. Figure 1 illustrates an example of the kind of SFPs used in this work. They can include membership functions of several shapes: triangular, trapezoidal, and semi-trapezoidal (only in the edges). One parameter C_i characterizes each fuzzy set A_i, except for the trapezoidal membership functions where two parameters, C_{i1} and C_{i2}, have to be considered. The optimization procedure will move the C_i points of each partition, given as a result new $C_i^{'}$ points that define a new SFP, without any ambiguity. The initial number and order of linguistic terms are maintained. This way, a very compact representation is got for the optimization procedure, while the SFP property is always kept.

Notice that there are other coding schemes, for instance [5] makes a very similar proposal considering two parameters for each fuzzy set disregarding its membership function shape. Thus, a vector of $2M$ real numbers characterizes a partition of M labels. As a result, the SFP property is kept, but not the membership function shapes. For example, a triangular function can derive to a trapezoidal one. We prefer to maintain at least the basic shape, even though the slopes can change, because it is strongly related to the linguistic term meaning. Remind that an expert supervises the fuzzy partition design and we do not want to lose the expert knowledge in the optimization phase. Other proposals like [9] code every characteristic point of the fuzzy sets, which gives more freedom degrees to the optimization but disregarding the SFP property. The same stands for recently proposed, advanced genetic tuning mechanisms such as [1] and [8]. Paper [12] shows how breaking the SFP property can yield more accurate systems, but at the cost of a loss of interpretability. This work illustrates that it is possible to achieve a good interpretability-accuracy trade-off through keeping the SFP property along the entire process.

Some authors [9] suggest the use of short variation intervals (T_i in figure 1) for each membership function parameter to prevent meaningful fuzzy sets. They are defined from the cross points between the adjacent fuzzy sets A_i and A_{i+1} in the initial partitions. As a consequence, the semantic consistency checking of the new partition is quite straightforward. Nevertheless, this constraint reduces significantly the search space and it makes more difficult to find a good solution. Therefore, this work also tries the use of extended variation intervals ($T_i^{'}$). In this case, each new $C_i^{'}$ point must be included between both the precedent (C_{i-1}) and the following (C_{i+1}) fuzzy set centers. In the edges, U_l ($i = 0$) and U_u ($i = M$) are considered. Thus, these two approaches are considered in the experiments: optimization constrained by short variation intervals (SI), and free optimization with extended variation intervals (EI).

In the following, the optimization algorithms under analysis, Solis-Wetts and Genetic Tuning, are deeply described. Firstly, the indices used for measuring KB accuracy, as well as the fitness function, are introduced.

A. KB accuracy

The two following indices are used to assess classification system accuracy:

- **Unclassified cases** (*UC*): Number of cases from the data set that do not fire at least one rule with a degree higher than Δ. In the experiments, Δ is equal to 0.1.
- **Error cases** (*EC*): Number of remaining cases for which observed and inferred values are different.

These indices convey complementary information. A good KB should minimize them by offering an accurate (reducing *EC*), and complete (reducing *UC*) set of rules. They can be combined to define the accuracy index:

$$Accuracy = 1 - \frac{EC + UC}{AC} \tag{3}$$

where *AC* stands for all cases in the data set. The goal of the optimization procedure is to maximize this accuracy index. In order to do that, the next fitness function is minimized. In the experiments, the value $a = 0.5$ is considered.

$$Fitness = a \cdot \frac{EC}{AC} + (1 - a) \cdot \frac{UC}{AC} \tag{4}$$

Let us now explain the algorithms used by each optimization strategy.

B. Solis-Wetts

System variables are ordered regarding the number of times they are used in the rule base. The procedure begins to optimize the most used variable. The detailed algorithm is described in [13]. Its pseudo-code is as follows:

1) Choose an initial vector of parameters to optimize, $C^{(0)}$. Initialize $S^{(0)} = 0$ and $k = 0$. S is a bias vector to memorize the previous successes.
2) Compute *Fitness*$(C^{(k)})$. Generate a Gaussian vector, $G^{(k)}$, with mean $S^{(k)}$. $G^{(k)} = S^{(k)} + N(0, \sigma)$.
3) **if** *Fitness*$(C^{(k)} + G^{(k)}) <$ *Fitness*$(C^{(k)})$
   then $C^{(k+1)} = C^{(k)} + G^{(k)}$
   $S^{(k+1)} = 0.4 \cdot G^{(k)} + 0.2 \cdot S^{(k)}$
   else if *Fitness*$(C^{(k)} - G^{(k)}) <$ *Fitness*$(C^{(k)})$
   then $C^{(k+1)} = C^{(k)} - G^{(k)}$
   $S^{(k+1)} = S^{(k)} - 0.4 \cdot G^{(k)}$
   else $C^{(k+1)} = C^{(k)}$
   $S^{(k+1)} = 0.5 \cdot S^{(k)}$
4) **If** $k >$ *MaxIter* **or** *Fitness* $<$ *StopThres*
   then stop
   else *k=k+1*; go to 2.

The algorithm stops when it gets the maximum number of iterations (*MaxIter*), or the fitness function (*Fitness*) is under a predefined threshold (*StopThres*). This procedure is repeated for each fuzzy partition. Two cases are studied:

- *Variable by Variable* (SW-V): Vector C includes all the C_i fuzzy set centers of the current partition.
- *Label by Label* (SW-L): The procedure is repeated for each linguistic term. Vector C includes only one C_i (one or two parameters) each time.

When a KB element (label or variable) is modified, the process comes back to the starting point (first variable to optimize; and first label in the partition in the case of SW-L). The procedure can be repeated for each KB element up to five times.

C. Genetic Tuning

The composition of the genetic tuning procedure considered is adapted from the proposal in [9]. The initial KB is used for building the first individual of the population. A real-coded chromosome is generated by joining the basic parameters C_i of every fuzzy partition. The variation interval T_i for each parameter C_i is also computed. Each basic parameter C_i is considered as a gene. The total number of genes is computed as the sum of the number of basic parameters by input variable. The rest of the population is randomly generated. A random value is assigned to each gene within its variation interval.

The pseudo-code is as follows:

1) Initialize the generation counter, $n = 0$.
2) Evaluate the initial population, $P^{(0)}$. Compute *Fitness* for each individual in the population.
3) **while** $n <$ *MaxGener* **and** *Fitness* $>$ *StopThres*
   n:=n+1
   Select $P^{(n)}$ from $P^{(n-1)}$
   Crossover $P^{(n)}$
   Mutate $P^{(n)}$
   Elitist selection
   Evaluate $P^{(n)}$
   end while

For each generation, the following steps are repeated:

- The selection of $P^{(n)}$ from $P^{(n-1)}$ is made as a deterministic tournament selection procedure. Each individual in the new population, $P^{(n)}$, is chosen from the old one, $P^{(n-1)}$, after making a tournament that involves *N* individuals randomly selected from $P^{(n-1)}$. The selection pressure can be adjusted by changing the tournament size, *N*. The best individual is selected in any tournament. The larger the value of *N*, the smaller the chances of weak individuals to be selected. For instance, if *N* is equal to the population length, then all individuals in $P^{(n)}$ are equal to the best one in $P^{(n-1)}$.
- A $BLX - \alpha$ crossover operator [11] is applied to $P^{(n)}$. Chromosomes of the current population, parents, are crossed over in pairs. Each pair of parents, $dad = (d_1, \cdots, d_g)$ and $mom = (m_1, \cdots, m_g)$, is substituted by two offspring, $O_d = (o_{d1}, \cdots, o_{dg})$ and $O_m = (o_{m1}, \cdots, o_{mg})$, where o_{dj} and o_{mj} are random values from the intervals [min_{dj}, max_{dj}] and [min_{mj}, max_{mj}], respectively. T_j=[T_j^l, T_j^u] is the variation interval of gene j.
  - min_{dj} = *maximum* $(T_j^l, d_j - \alpha \cdot |d_j - m_j|)$
  - max_{dj} = *minimum* $(d_j + \alpha \cdot |d_j - m_j|, T_j^u)$
  - min_{mj} = *maximum* $(T_j^l, m_j - \alpha \cdot |m_j - d_j|)$
  - max_{mj} = *minimum* $(m_j + \alpha \cdot |m_j - d_j|, T_j^u)$
- A uniform mutation operator is considered. The value of the selected gene is changed by other one randomly generated within its variation interval.

TABLE I
GENETIC TUNING CONFIGURATION PARAMETERS.

Population length	60
Tournament size (N)	2
Mutation probability	0.1
Crossover probability	0.6
α-crossover	0.3

- Finally, the elitist selection ensures the selection of the best individual of the previous generation.

The procedure stops when it gets the maximum number of generations (*MaxGener*), or *Fitness* is under the predefined threshold (*StopThres*). In the experiments, *StopThres* is equal to zero. The rest of parameters are detailed in table I.

IV. RESULTS AND DISCUSSION

The two optimization strategies proposed in this paper have been evaluated using the well-known benchmark classification problem WBCD (Wisconsin breast cancer). This database[1] was obtained from the University of Wisconsin Hospitals, Madison, from Dr. William H. Wolberg. It consists of 683 samples (incomplete patterns with missing values are not taken into consideration) that involve 9 features obtained from fine needle aspirates, for two cancer states (benign or malignant). WBCD is a medical diagnosis problem. In this kind of application the KB interpretability-accuracy trade-off is of prime importance.

First of all, HILK methodology[2] was used for building a compact KB, with a simultaneous good trade-off regarding training and test patterns. A 5-fold cross-validation[3] is made over the whole data set. It is divided into 5 parts of equal size, and each part keeps the original distribution (percentage of elements for each class) in the whole set. Table II describes the KB basic parameters and the accuracy index averaged over the five folds. Notice that we have selected the *Minimum t-norm* as conjunctive operator, and the *winner rule* fuzzy reasoning mechanism.

The well-known Quinlan's C4.5 algorithm, introduced in [20], has been selected as comparison baseline because it builds decision trees which are acknowledged as a very interpretable knowledge representation. Nevertheless, they are crisp trees, and as a result it is not considered as a robust technique because their accuracy strongly depends on the crisp threshold values that define their configuration. Interpretability is assessed in terms of tree dimension (number of leaves and tree size). In order to make a comparison with HILK, the number of leaves can be compared to the total number of rules, and the tree size (computed as the sum of the number of nodes in every branch) is equivalent to

[1]The data set is available from the UCI machine learning repository (http://www.ics.uci.edu/~mlearn/MLSummary.html)

[2]Let us remark that current contribution is not dedicated to explain the entire methodology but only the final optimization phase. Please refer to the cited literature ([3] and [16]) for a deeper description.

[3]Cross-validation is a method for estimating generalization error based on resampling [17]. It is often used for choosing among different models.

TABLE II

KB CONFIGURATION AND ACCURACY (C4.5 VS. HILK).

Parameters	C4.5		HILK	
	Mean	Standard deviation	Mean	Standard deviation
Number of rules	11.4	1.9493	3.6	0.5477
Number of premises	46.8	13.5166	5.6	1.5166
Number of input variables	5.6	1.1402	2.6	0.5477
Accuracy index (training)	0.9821	0.0039	0.944	0.0202
Accuracy index (test)	0.9546	0.0131	0.9384	0.0319

the total number of premises. Table II shows the KB basic parameters and the accuracy index averaged with C4.5. Note that we have used the implementation of C4.5 in Weka [24], a free software tool for data mining tasks, with the same 5 fold sets used by HILK.

The comparison between HILK and C4.5 in table II lets us draw some conclusions. HILK yields more interpretable KBs, with a smaller number of premises and rules. Note that the number of inputs is clearly smaller than the initial one (9). However, C4.5 achieves more accurate KBs. As accuracy and interpretability are conflicting goals, we can argue that interpretability improvement is obtained at the cost of a loss of accuracy. Therefore, it seems reasonable to make an optimization of KBs obtained by HILK, in order to get a better interpretability-accuracy trade-off. The two optimization procedures presented in this work have been applied to these KBs, with the aim of improving their accuracy indices while keeping their high interpretability.

Table III shows the main results. The first column shows the name of the method used for building the initial KBs, followed by the optimization strategy, and in brackets the type of variation intervals and also a number relative to *MaxIter* or *MaxGener* depending on the optimization algorithm. *SW-V* stands for *Solis-Wetts Variable by Variable*, *SW-L* means *Solis-Wetts Label by Label*, and *GT* is *Genetic Tuning*. Each strategy is evaluated with *SI* (short variation intervals) and *EI* (extended variation intervals). The last column shows the mean time in seconds spent by the runs (on a Pentium IV 1.8 GHz and 1 GB RAM). The other columns show the accuracy index over training and test sets, using the arithmetic mean and the standard deviation. C4.5 and HILK accuracy indices are included in this table for making easier the comparison with the optimization results. They are obtained through 5-fold cross-validation. However, six runs for each fold are made in order to assess the optimization strategies random nature. Therefore, the mean and standard deviation values are computed over 30 different runs of each method.

HILK optimization results are quite similar for both strategies (*SW* and *GT*). There is an accuracy improvement regarding both training and test patterns, but it is larger over test ones. Although this improvement is not very significant, we are able to get a much simpler (and thus much more interpretable) fuzzy classifier with a test classification error only less than one percent higher than that of C4.5. We should remark that a larger accuracy increase could be obtained in case of relaxing the SFP property but we prefer to keep it in order to maintain the comprehensibility of the KB as high as possible.

GT yields the best results. The larger the value of *MaxGener* the higher the accuracy. Besides, *SW-L* achieves higher accuracy than *SW-V*. *SW* results are slightly better considering *EI*, but there is no change regarding *MaxIter*. This is due to our iterative application of the algorithm.

In order to check thoroughly the effect of the variation intervals (*SI* or *EI*), we have built HILK-REG which corresponds to the same KBs built by HILK but changing the automatically learnt fuzzy partitions for uniformly defined ones keeping the same number of linguistic terms. Consequently, HILK-REG partitions are worse fitted than HILK ones, so their accuracy is smaller. HILK-REG optimization is clearly better for *GT* and *EI*. We can conclude that *GT* yields similar results no matter the initial KB (HILK or HILK-REG), but *SW* achieves more accurate results starting from HILK. On the one hand, if a suited solution is taken as starting point, then a local search strategy like *SW* is able to yield very good results in a few iterations. On the other hand, if the initial solution is not so good, a global search strategy like *GT* seems much more effective. Finally, the use of *EI* spreads the search space and lets us achieve more accurate solutions. Meaningful fuzzy sets are guaranteed through keeping the SFP property. However, it should be notice that the use of *EI* could lead to change the meaning of the initial fuzzy sets.

Lastly, *SW* is much more efficient than *GT* regarding computing time. *SW* only spends a few seconds by run while *GT* spends a few minutes. GT yields greater computational cost due to the evolutionary process that involves the evaluation of the entire population for each generation.

V. CONCLUSIONS

This paper deals with the interpretability-accuracy trade-off paradigm. It shows how it is possible to build highly interpretable KBs using linguistic variables with SFPs and linguistic rules. Fuzzy modeling based on using SFPs favours interpretability but it penalizes accuracy due to it is a very strong constraint. However, the use of optimization strategies lets us improve accuracy. As a result, we are able to get a good trade-off between both modeling criteria.

In the context of HILK methodology, the optimization process starts from a KB that gives us a quite good solution regarding accuracy and interpretability. Therefore, the use

TABLE III

OPTIMIZATION AVERAGED RESULTS (6×5-FOLD CROSS-VALIDATION).

Method	Accuracy				Mean Run time (seconds)
	Training		Test		
	Mean	Standard deviation	Mean	Standard deviation	
C4.5	0.9821	0.0039	0.9546	0.0131	-
HILK	0.944	0.0202	0.9384	0.0319	-
HILK + SW-V (SI, 10)	0.9462	0.022	0.9428	0.0351	1.2
HILK + SW-V (EI, 10)	0.9462	0.022	0.9428	0.0351	1.2
HILK + SW-L (SI, 10)	0.9477	0.0222	0.9443	0.0332	2
HILK + SW-L (EI, 10)	0.948	0.0226	0.9443	0.0332	2
HILK + GT (SI, 100)	0.9474	0.02	0.9443	0.0293	117.5
HILK + GT (EI, 100)	0.9472	0.02	0.9445	0.0291	120.5
HILK + GT (SI, 1000)	0.948	0.0206	0.9465	0.0297	344.3
HILK + GT (EI, 1000)	0.9483	0.0207	0.9462	0.0299	357.2
HILK-REG	0.8723	0.0568	0.8739	0.0841	-
HILK-REG + SW-L (SI, 10)	0.9191	0.0502	0.9193	0.0638	5
HILK-REG + SW-L (EI, 10)	0.937	0.0171	0.9443	0.0324	3.8
HILK-REG + GT (SI, 1000)	0.9231	0.0335	0.9135	0.0674	333.9
HILK-REG + GT (EI, 1000)	0.9483	0.0207	0.947	0.0307	334.8

of the SW-L strategy seems to be the best option for the current data set if the run time is a key concern. It increases the accuracy in a short run time. Otherwise, the GA gives a more accurate classifier for both the training and test sets.

All results presented in this paper were reached using KBCT [4], a free software tool (distributed under the terms of the GNU General Public License) for generating and refining fuzzy knowledge bases.

REFERENCES

[1] R. Alcalá, J. Alcalá-Fdez, M. J. Gacto, and F. Herrera, "Rule base reduction and genetic tuning of fuzzy systems based on the linguistic 3-tuples representation," *Soft Computing*, vol. 11(5), pp. 401–419, 2007.

[2] R. Alcalá, J. Alcalá-Fdez, J. Casillas, O. Cordón, and F. Herrera, "Hybrid learning methods to get the interpretability-accuracy trade-off in fuzzy modeling," *Soft Computing*, vol. 10(9), pp. 717–734, 2006.

[3] J. M. Alonso, L. Magdalena, and S. Guillaume, "Linguistic knowledge base simplification regarding accuracy and interpretability," *Mathware & Soft Computing*, vol. 3, 2006.

[4] J. M. Alonso, L. Magdalena, and S. Guillaume, "KBCT: A knowledge management tool for fuzzy inference systems.," *Free software under GPL license, available in http://www.mat.upm.es/projects/advocate/kbct.htm*, 2003.

[5] E. V. Broekhoven, V. Adriaenssens and B. De Baets, "Interpretability-preserving genetic optimization of linguistic terms in fuzzy models for fuzzy ordered classification: An ecological case study," *Int. Journal of Approximate Reasoning*, vol. 44, pp. 65–90, 2007.

[6] J. Casillas, O. Cordón, F. Herrera, and L. Magdalena (Eds.), "Accuracy improvements in linguistic fuzzy modeling," *Studies in Fuzziness and Soft Computing, Springer-Verlag, Heidelberg*, vol. 129, 2003.

[7] J. Casillas, O. Cordón, F. Herrera, and L. Magdalena (Eds.), "Interpretability issues in fuzzy modeling," *Studies in Fuzziness and Soft Computing, Springer-Verlag, Heidelberg*, vol. 128, 2003.

[8] J. Casillas, O. Cordón, M. J. del Jesús, and F. Herrera, "Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction," IEEE Trans. on Fuzzy Systems, vol. 13(1), pp. 13–29, 2005.

[9] O. Cordón and F. Herrera, "A three-stage evolutionary process for learning descriptive and approximate fuzzy logic controller knowledge bases from examples," *Int. Journal of Approximate Reasoning*, vol. 17(4), pp. 369–407, 1997.

[10] O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena, "Genetic Fuzzy Systems: Evolutionary tuning and learning of fuzzy knowledge bases," *Adv. in Fuzzy Systems - Applications and Theory.*, vol. 19, 2001.

[11] L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithms and interval schema," *Foundations of Genetic Algorithms 2*, L. D. Whitley, editor, pp. 185–202, 1993.

[12] P. Fazendeiro and J. Valente de Oliveira, "A working hypothesis on the semantics/accuracy synergy," *Proc. of the Joint EUSFLAT-LFA 2005 conference*, Barcelona, Spain, pp. 266–271, 2005.

[13] P.-Y. Glorennec, "Algorithmes d′apprentissage pour systèmes d′inférence floue, (in French)" *Editions Hermès, Paris*, 1999.

[14] D. E. Goldberg, "Genetic algorithms in search, optimization, and machine learning," *Addison-Wesley, New York*, 1989.

[15] S. Guillaume, "Designing fuzzy inference systems from data: an interpretability-oriented review," *IEEE Trans. on Fuzzy Systems*, vol. 9(3), pp. 426–443, 2001.

[16] S. Guillaume and L. Magdalena, "Expert guided integration of induced knowledge into a fuzzy knowledge base," *Soft Computing*, vol. 10(9), pp. 773–784, 2006.

[17] J. S. U. Hjorth, "Computer Intensive Statistical Methods Validation, Model Selection, and Bootstrap," *Chapman & Hall, London*, 1994.

[18] E. H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic systems," *IEEE Trans. on Computers*, vol. 26(12), pp. 1182–1191, 1977.

[19] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *The Psychological Review*, vol. 63(2), pp. 81–97, 1956.

[20] J. R. Quinlan, "C4.5: Programs for machine learning," *Morgan Kaufmann Publishers, San Mateo, CA*, 1993.

[21] E. H. Ruspini, "A new approach to clustering," *Information and Control*, vol. 15(1), pp. 22–32, 1969.

[22] F. Solis and J. Wetts, "Minimization by random search techniques," *Mathematics of Operation Research*, vol. 6, 1981.

[23] J. Valente de Oliveira, "Semantic constraints for membership function optimization," *IEEE Trans. on Systems, Man and Cybernetics. Part A, Systems and Humans*, vol. 29(1), pp. 128–138, 1999.

[24] I. H. Witten and E. Frank, "Data Mining: Practical machine learning tools and techniques," *2nd Edition, Morgan Kaufmann*, 2005.

[25] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[26] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning," *Parts I, II, and III. Information Sciences*, vol. 8, 8, 9, pp. 199–249, 301–357, 43–80, 1975.