

TUNING THE PARAMETERS OF A CLASSIFIER FOR FAULT DIAGNOSIS

Particle Swarm Optimization vs Genetic Algorithms

Cosmin Danut Bocaniala

Computer Science and Engineering Department, "Dunarea de Jos" University, Domneasca 47, Galati, Romania

José Sa da Costa

GCAR/IDMEC, Department of Mechanical Engineering, Instituto Superior Tecnico, Technical University of Lisbon, Lisbon, Portugal

Keywords: Particle swarm optimization, Parameters, Fault diagnosis, Pattern recognition, Fuzzy logic

Abstract: This paper presents a comparison between the use of particle swarm optimization and the use of genetic algorithms for tuning the parameters of a novel fuzzy classifier. In the previous work on the classifier, the large amount of time needed by genetic algorithms has been significantly diminished by using an optimized initial population. Even with this improvement, the time spent on tuning the parameters is still very large. The present comparison suggests that using particle swarm optimization may improve considerably the time needed for tuning the parameters. This way, the fuzzy classifier becomes suitable for real world application. The result is validated by application to a fault diagnosis benchmark.

1 INTRODUCTION

A fault diagnosis system is a monitoring system that is used to detect faults and diagnose their location and significance in a system (Chen and Patton, 1999). The diagnosis system performs mainly the following tasks: fault detection – to indicate if a fault occurred or not in the system, and fault isolation – to determine the location of the fault. One of the main perspectives on fault diagnosis is to consider it a classification problem (Leonhardt and Ayoubi, 1997). The symptoms are extracted on the basis of the measurements provided by the actuators and sensors in the monitored system. The actual diagnostic task is to map data points the symptoms space into the set of considered faults.

The research literature offers three possible directions to develop fuzzy classifiers for fault diagnosis: mixtures of neural networks and fuzzy rules (Calado *et al.*, 2001; Palade *et al.* 2002), sets of fuzzy rules that describe the relationships symptoms-faults using transparent linguistic terms (Frank, 1996; Koscielny *et al.*, 1999), and collections of fuzzy subsets that represent the normal state and each faulty state of the system (Boudaoud,

2000). The fuzzy classifier addressed in this paper follows the third direction and it was proposed in (Bocaniala, 2003; Bocaniala and Sa da Costa, 2003). The main advantages of the classifier are the high accuracy with which it delimits the areas corresponding to different categories, and the fine precision of discrimination inside overlapping areas. In the previous work, the parameters of the classifier have been tuned using genetic algorithms. Even if the large amount of time needed by genetic algorithms has been significantly diminished by using an optimized initial population, the time spent on tuning the parameters is still too large.

This paper presents a comparison between the use of particle swarm optimization (PSO) and the use of genetic algorithms for tuning the parameters of a novel fuzzy classifier. The present comparison suggests that using particle swarm optimization may improve considerably the time needed for tuning the parameters. This way, the fuzzy classifier becomes suitable for real world application. The result is validated by application to a fault diagnosis benchmark.

However, there is no other fault diagnosis related benchmark that may have permitted a comparison

between the performances of the fuzzy classifier on different benchmarks. Though, a comparative study on the performance of the fuzzy classifier when applied on different data sets is given in (Bocaniala, 2003).

The paper is structured as follows. Section 2 presents the main theoretical aspects of the fuzzy classifier. Section 3 briefly describes the PSO technique and the variant used in this paper. Next section, Section 4, introduces the case study, the DAMADICS benchmark (<http://www.eng.hull.ac.uk/research/control/damadics1.htm>). Section 5 presents the comparison between the performance of the classifier when genetic algorithms and respectively PSO are used. Finally, some conclusions are drawn and further research directions are identified.

2 THE FUZZY CLASSIFIER

The fuzzy classifier addressed in this paper has been recently introduced by (Bocanialac, 2003; Bocaniala and Sa da Costa, 2003). The classifier relies on the use of a similarity measure between points in the space associated to the problem. The first subsection presents the way similarity measures are used to induce the fuzzy sets associated with each category. The second subsection describes the management of the available data in order to design and to test the classifier.

2.1 Fuzzy sets induced by a measure of similarity

The classifier performs its task using a measure of similarity between points in the space associated with the problem. The similarity of data points within the same category is larger than the similarity of data points belonging to different categories. The similarity between two points u and v , $s(u,v)$, will be expressed using a complementary function, $d(u,v)$, expressing dissimilarity. The dissimilarity measure is encoded via function $h^\beta(\delta(u,v))$ that depends on one parameter, β , and that maps the distance between u and v , $\delta(u,v)$, into $[0,1]$ interval (Eq. 1). The maximum value for $d(u,v)$, which is equal to $h^\beta(\delta(u,v))$, is 1. It follows that the functions s and d are complementary with regard to this value; thus, $s(u,v)=1-d(u,v)$.

The similarity measure between two data points can be extended to a similarity measure between a data point and a subset of data points. The similarity between a data point u and a subset is called the subset affinity measure. Let $C=\{C_i\}_{i=1,\dots,m}$ be the

partition of a set of data points according to the category they belong to. The subset affinity measure between a data point u and a category C_i is given by Eq. 2, where n_i denotes the number of elements in C_i .

$$h^\beta(\delta(u,v)) = \begin{cases} \delta(u,v)/\beta, & \text{for } \delta(u,v) \leq \beta \\ 1, & \text{otherwise} \end{cases} \quad (1)$$

$$r(u,C_i) = \frac{\sum_{v \in C_i} s(u,v)}{n_i} = 1 - \frac{1}{n_i} \sum_{v \in C_i} h^\beta(\delta(u,v)) \quad (2)$$

The effect of using the β parameter is that only those data points from C_i , whose distance to u is larger than β , contribute to the affinity value. The explanation is that only these points have a non-zero similarity with u . It follows that the affinity of a data point u with different categories in the partition is decided within the neighbourhood defined by β .

The natural belongingness of a data point to a category varies between a maximum value and a minimum value (corresponding to no belongingness) and it can be approximated using the subset affinity measure. Therefore, each category C_i (which represents a classical set) is replaced by a fuzzy set because the belongingness to this type of sets varies inside $[0,1]$ interval. The fuzzy sets are induced by the corresponding categories as denoted by Eq. 3. The term $r(u,C)$ expresses the affinity of u to the whole set C , the n value represents the cardinal of C , and the n_i value represents the cardinal of C_i .

$$\mu_i(u) = \frac{r(u,C_i)}{r(u,C)} = \frac{n_i - \sum_{v \in C_i} h^\beta(\delta(u,v))}{n - \sum_{v \in C} h^\beta(\delta(u,v))} \quad (3)$$

Using only one similarity measure does not always provide satisfactory results (Bocaniala, 2003; Bocaniala and Sa da Costa, 2003). Thus, the advantages brought by two or more similarity measures may be combined in order to improve the performance of the classifier, i.e. a hybrid approach may be used (Bocaniala, Sa da Costa and Palade, 2004). In this paper, it will be used a hybrid approach based on Euclidean distance and Pearson correlation (Weisstein, 1999). The β parameter will be applied only to the similarity measure induced by the Euclidean distance. Two subset affinity measures are used, based on the two similarity measures induced by the Euclidean distance and respectively Pearson correlation. Finally, the fuzzy membership

functions will be combinations of the two subset affinity measures.

2.2 Classification based on induced fuzzy sets

Let m be the number of the categories considered for the problem to be solved. First the set of all available data C is partitioned according to the category to which each data point belongs. The partition is formed by the subsets C_i , $i=1, \dots, m$. In order to design and test the classifier, each set C_i is split in three representative and distinct subsets, C_i^{ref} , C_i^{param} and C_i^{test} . On the basis of these subsets, three unions, REF , $PARAM$, and $TEST$, are defined (Eq. 4). They are called the reference patterns set, the parameters tuning set and respectively the test set.

$$\begin{aligned} REF &= \bigcup_{i=1}^m C_i^{ref} \\ PARAM &= \bigcup_{i=1}^m C_i^{param} \quad (4) \\ TEST &= \bigcup_{i=1}^m C_i^{test} \end{aligned}$$

A subset is considered representative for a given set if it covers that set in a satisfactory manner. The semantic for the expression satisfactory covering subset adopted in this paper is that such a set represents a subset of data that preserves (with a given order of magnitude) the distribution of the data associated to the problem. Selecting the elements that compose a satisfactory covering subset for a given data set can be costly. Therefore, it is more convenient to use a selection method that provides convenient approximations for satisfactory covering subsets. Such a method is proposed in (Bocaniala, 2003).

In the following, the role of each one of the previous three unions is detailed. It is to be noticed that the union of subsets having the satisfactory covering property for a set represents also a satisfactory covering subset of that set.

1.1.1 The REF set

The subset affinity measures are defined for the representative subsets C_i^{ref} , $i=1, \dots, m$. Notice that the affinity measures differ from a representative subset to another as they depend on different β_i parameters. The practice showed that using different parameters for different categories increases substantially the performance of the classifier. Using these affinity measures, Eq. 5 defines the induced fuzzy sets $Fuzz_i$.

$$\mu_i(u) = \frac{n_i - \sum_{v \in C_i^{ref}} h^{\beta_i}(\delta(u, v))}{n - \sum_{v \in \bigcup_{i=1}^m C_i^{ref}} h^{\beta_i}(\delta(u, v))} \quad (5)$$

An object u presented at the input of the classifier is assigned to the category C_z whose corresponding degree of assignment $\mu_z(u)$ is the largest (Eq. 6). In case of ties, the assignment to a category cannot be decided and the object is rejected.

$$u \in z\text{-th category} \Leftrightarrow \mu_z(u) = \max_{i=1, \dots, m} \mu_i(u) \quad (6)$$

1.1.2 The PARAM set

The shape of the membership functions μ_i , associated to the fuzzy sets $Fuzz_i$, depends on the representative subset C_i^{ref} but also on the value of β_i parameter, $i=1, \dots, m$ (Eq. 5). The algorithm for tuning the parameters β_i of the classifier represents a search process in a m -dimensional space for the parameters vector $(\beta_1, \beta_2, \dots, \beta_m)$ that meets, for each category, maximal correct classification criteria and minimal misclassification criteria.

Previous work performs this search with the help of genetic algorithms that start from an optimized initial population (Bocaniala, 2003; Bocaniala and Sa da Costa, 2003). The fitness of an individual from the population is given by the degree with which the associated parameters accomplish the two mentioned criteria. In order to approximate the degree of accomplishment, the performance of the classifier when applied to $PARAM$ set is used. Since the $PARAM$ set represents a satisfactory covering set for the set of all available data, the performance of the classifier on this set represents an approximation of the performance of the classifier on the set of all possible data associated to the problem.

1.1.3 The TEST set

The performance of the classifier is measured according to its generalization capabilities when applied on the $TEST$ set. The practice showed that the performance of the classifier may improve if the testing is performed after adding the data in the $PARAM$ set to the REF set.

2 PARTICLE SWARM OPTIMIZATION

The PSO methodology has been recently introduced in the field of Evolutionary Computing by (Kennedy and Eberhart, 1995). The main idea is to use mechanisms found by studying the flight behaviour of bird flocks (Heppner and Grenander, 1990). The method may be used to solve optimization problems using the next analogy. If a roosting area is set, then the birds will form flocks and will fly towards this area, "landing" when they arrived there. The roosting area may be seen as an optimal or a near-optimal solution in the search space. The birds may represent points in the search space that will move in time towards this solution. The search process is guided by an objective function and each point is able to evaluate the value of this function (the fitness) for its current location. The movements of the points during search will no longer resemble the move of the birds in a flock, but rather the movement of the particles in a swarm. There are two mechanisms that are employed during this exploration of the search space. First, each point in the swarm memorizes the best location (in terms of fitness) he ever passed through. Second, each point is aware of the best location that the whole swarm ever passed through, i.e. the global best location. The new location of a particle is computed as follows. Using the vector notation from Physics, the direction vector of each particle is updated using the vectors that point from the current location towards the two previously mentioned locations. The search process stops when all other points draw closer than a very small given distance to one point. This point is considered to be the solution of the optimization problem.

The variant of PSO used in this paper starts from a set of points around origin in the parameters m -dimensional space mentioned in Subsection 2.1.3. Fortunately, the probability to find points with large fitness around origin is very high (see Table 1). This means that it is very likely that the search process starts with particles found very close to optimal solutions. The exploration of the search space follows the rules discussed above. The stop condition is modified as follows. It was noticed that if the global best location does not modify for a relatively small number of iterations then this location is a optimal solution. On the basis of this observation, the search process will be stopped if the global best location does not change for 3 iterations. Using the analogy above, if the roosting is found then the global best location will not further modify and, therefore, there is no reason to wait until all the birds landed.

3 CASE STUDY

The DAMADICS benchmark flow control valve was chosen as the case study for this method. More information on DAMADICS benchmark is available via web, <http://www.eng.hull.ac.uk/research/control/damadics1.htm>. The valve was extensively modeled and a MATLAB/SIMULINK program was developed for simulation purposes (Sa da Costa and Louro, 2003). The data relative to the behavior of the system while undergoing a fault was generated using as inputs to the simulation real data, normal behavior and some faulty conditions, collected at the plant. This method provides more realistic conditions for generating the behavior of the system while undergoing a fault. It also makes the FDI task more difficult because the real inputs cause the system to feature the same noise conditions as those in the real plant. However the resulting FDI systems will deal better when applied to the real plant.

The system is affected by a total of 19 faults. In this paper only the abrupt manifestation of the faults has been considered. A complete description of the faults and the way they affect the valve can be found in (Louro, 2003). There are several sensors included in the system that measure variables that influence the system, namely the upstream and downstream water pressures, the water temperature, the position of the rod, and the flow through the valve. These measurements are intended for controlling the process but they can also be used for diagnosis purposes, which means that the implementation of this sort of system will not imply additional hardware. Two of these sensors, the sensor that measures the rod position (x) and the sensor that measures the flow (F) provide variables that contain information relative to the faults. The difference dP between the upstream pressure ($P1$) sensor measurement and the downstream pressure ($P2$) sensor measurement is also considered (besides F and x) as it permits to differentiate $F17$ from the other faults. For the rest of the faults, the previous difference has always negligible values (close to zero).

The effects of six out of the 19 faults on this set of sensor measurements are not distinguishable from the normal behaviour, $\{F4, F5, F8, F9, F12, F14\}$. So, in the following, these cases are not studied. They can be dealt with if further sensors are added to the system. Also, there can be distinguished three groups of faults, $\{F3, F6\}$, $\{F7, F10\}$, and $\{F11, F15, F16\}$, that share similar effects on the measurements. Due to the large overlapping, a fault member in one of the previous groups can be easily mistaken with faults in the same group. This problem is solved in recent studies by using a hybrid

similarity measure based on Euclidean distance and Pearson correlation in order to distinguish between elements in the previous three groups of faults.

4 PARTICLE SWARM OPTIMIZATION VS GENETIC ALGORITHMS

The 13 faults distinguishable by the normal state were simulated two times for 20 values of fault strength, uniformly distributed between 5% and 100%, and different conditions for the reference signal. The strength of a fault represents the intensity with which the fault acts on the valve. Generally, for small to medium fault strengths, the effects of the faults on the valve are not distinguishable from the normal state. The previous settings approximate very well all possible faulty situations involving the 13 faults. The data obtained during the first simulation have been used to design the classifier, i.e. 50% for the *REF* set, 50% for the *PARAM* set. The data obtained during the second simulation have been used as the *TEST* set.

The objective function used in previous work (Bocaniála, 2003) with genetic algorithms is also used with PSO. This objective function computes the fitness of a set of parameters using the *confusion matrix* obtained when applying the classifier on the *PARAM* set. The fitness represents a weighted sum of all elements in this matrix. Each element on the main diagonal represents the percent of well-classified data for that category and is weighted by $m - 1$ – the number of categories considered. An element not member of the main diagonal, found on row i and column j , $i \neq j$, represents the percent of data from the i -th category misclassified as belonging to the j -th category. These elements are weighted by -1 . Notice that the objective function encourages mainly the growth of percentage of the well-classified data while still penalising the misclassifications occurred. The maximum fitness is obtained when all data are correctly classified, i.e. the confusion matrix represents the identity matrix. In this case, the fitness value is m (the weight for elements on the main diagonal) $\times m$ (the length of the main diagonal). Given the fact that for our case study the value of m is 14 (one normal state and 13 faulty states), the maximum fitness that may be reached is 196.

As detailed in Subsection 2.2., the suitability of a set of parameters of the classifier is given by the performance of the classifier on the *PARAM* set. Thus, checking a set of parameters corresponds to one call of the classification procedure on the

PARAM set. The comparison between the hill-climbing technique and genetic algorithms has been performed by counting the number of calls of the classifier during the search process. The time spent for one call of the classifier is the same for both methodologies. The amount of time needed by one call of a classifier on a computer with Intel Pentium 4 at 2.4 Ghz, 526 MB RAM is 3 seconds. This large amount of time may be explained by the large size of the *REF* and *PARAM* sets.

The settings used for the genetic algorithm are next. Each population contains 20 individuals and only the first 20 successive generations are produced. The genetic algorithm always starts from an optimized initial population generated using the algorithm in (Bocaniála, 2003). For each new population, the best 3 individuals from the previous generation are kept and 2 new individuals are randomly generated. The settings used for the PSO method have been already discussed in Section 3. It is very important to notice that PSO does not use an optimized initial population. Though, it makes use of the fact that there is a high probability that the fitness of the initial particles is considerably large (see Section 3).

Table 1 Comparison between classifier performance when using genetic algorithm (GA) and particle swarm optimization (PSO) for parameters tuning

No. exp (Method)	Initial fitness	Final fitness	No. calls classifier
1 (GA)	138.83	147.98	340
2 (GA)	140.40	149.54	340
3 (GA)	144.97	151.26	340
4 (GA)	140.83	149.82	340
5 (GA)	139.98	151.08	340
1 (PSO)	124.18	151.25	100
2 (PSO)	121.62	150.03	220
3 (PSO)	117.65	146.46	160
4 (PSO)	127.47	151.38	140
5 (PSO)	134.07	156.30	140

Using the previous settings, five experiments have been performed for both methodologies. For each experiment, the next information is recorded: the maximum initial fitness (inside the initial optimized population and respectively inside the initial swarm), the maximum fitness reached, and the number of calls of the classifier. The results are shown in Table 1. Analysing the content of Table 1, two facts may be deduced. First, the initial maximum fitness for PSO is usually smaller than the one for the genetic algorithm, while the final maximum fitness for PSO is usually the same or

slightly larger than the one for the genetic algorithm. Second, the number of calls needed for PSO is from one third to two thirds less than the number of calls for the genetic algorithm. The conclusion is that using PSO methodology instead genetic algorithms provides better performance of the classifier but with a much lower cost in terms of number of calls of the classifier.

5 CONCLUSIONS

This paper presented a comparison between the use of particle swarm optimization and the use of genetic algorithms for tuning the parameters of a novel fuzzy classifier. The comparison suggests that using particle swarm optimization may improve considerably the time needed for tuning the parameters. The result is validated by application to a fault diagnosis benchmark that presents large overlapping between constituent categories, i.e. the normal state and the faulty states. The computational time needed by particle swarm optimisation is from one third to two thirds less than the time needed by genetic algorithms. Due to this improvement regarding the computational time the classifier becomes more suitable for application to fault diagnosis of real world systems.

REFERENCES

- Baker, E. (1978). *Cluster analysis by optimal decomposition of induced fuzzy sets (PhD thesis)*. Delftse Universitaire Pres, Delft, Holland.
- Bocaniala, C.D. (2003). *Tehnici de inteligență artificială aplicate în diagnoza defectelor: Aplicații ale tehnicilor de clasificare (Technical Research Report within doctoral training)*. University "Dunarea de Jos" of Galati, Romania. (available in English for download at <http://www.gcar.dem.ist.utl.pt/Pessoal/Cosmin/publicat.htm>)
- Bocaniala, C.D., J. Sa da Costa and R. Louro (2003). A Fuzzy Classification Solution for Fault Diagnosis of Valve Actuators. In: *Proceedings of the 7th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, Oxford, UK, September 3-5*, Part I, pp. 741-747. LNAI Series, Springer-Verlag, Heidelberg, Germany.
- Bocaniala, C. D., J. Sa da Costa and V. Palade (2004). A Novel Fuzzy Classification Solution for Fault Diagnosis, *International Journal of Fuzzy and Intelligent Systems*. (accepted)
- Boudaoud, N. and M. Masson (2000). Diagnosis of transient states using pattern recognition approach. *JESA – European Journal of Automation*, **34**, 689-708.
- Calado, J. M. G., J. Korbicz, K. Patan, R. Patton and J. M. G. Sa da Costa (2001). Soft Computing Approaches to Fault Diagnosis for Dynamic Systems. *European Journal of Control*, **7**, 248-286.
- Chen, J. and R. J. Patton (1999). *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Asian Studies in Computer Science and Information Science, Kluwer Academic Publishers, Boston, USA.
- European Community's FP5, Research Training Network DAMADICS Project, <http://www.eng.hull.ac.uk/research/control/damadics1.htm>.
- Frank, P.M. (1996). Analytical and qualitative model-based fault diagnosis – a survey and some new results. *European Journal of Control*, **2**, 6-28.
- Heppner, F. and Grenander U. (1990). A stochastic nonlinear model for coordinated bird flocks. In: *The Ubiquity of Chaos*, AAAS Publications, Washington, DC.
- Kennedy, J. and Eberhart, R. (1995). Particle Swarm Optimization, In: *Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia*.
- Koscielny, J.M., M. Syfert and M. Bartys (1999). Fuzzy-logic fault diagnosis of industrial process actuators. *International Journal of Applied mathematics and Computer Science*, **9**, 653-666.
- Leonhardt, S. and M. Ayoubi (1997) Methods of fault diagnosis. *Control Engineering Practice*, **5**, 683-692.
- Louro, R. (2003). *Fault Diagnosis of An Industrial Actuator valve (MSc dissertation)*. Technical University of Lisbon, Lisbon, Portugal.
- Palade, V., R. J. Patton, F. J. Uppal, J. Quevedo, S. Daley (2002). Fault diagnosis of an industrial gas turbine using neuro-fuzzy methods. In: *Preprints of the 15th IFAC World Congress, Barcelona, Spain*, CD-ROM.
- Sá da Costa, J. and R. Louro (2003). Modelling and simulation of an industrial actuator valve for fault diagnosis benchmark. In: *Proceedings of the Fourth International Symposium on Mathematical Modelling, Vienna, Austria*.
- Weisstein, E.W. (1999). Correlation Coefficient. From *MathWorld--A Wolfram Web Resource*, <http://mathworld.wolfram.com/CorrelationCoefficient.html>