Fault Diagnostics of Intelligent Pump

Jan Kout, Jiří Klema, Olga Štěpánková
Department of Cybernetics, Faculty of Electrical Engineering,
The Gerstner Laboratory, Czech Technical University,
Technicka 2, CZ 166 27 Praha 6, Czech Republic
email: {kout, klema, step}@labe.felk.cvut.cz

Abstract

This paper introduces a competence estimate approach to process fault diagnosis. The fault examples are extended by estimates of competence to individual classifications, which are used as additional attributes. Several machine learning (ML) techniques are applied on the training set of examples to develop a methodology which can generate competence estimates for unseen cases and finally suggest their classification in a crisp form. The features of the method are shown on a case study concerned with condition monitoring of induction motor driven pumps. The experiments indicate that the suggested method gives better results than fault prediction with aid of direct application of traditional ML techniques.

1 Introduction

Condition monitoring of induction motor driven pumps, involves detection of commonly encountered faults associated with the pumps before these faults become very serious (resulting e.g. in the shutdown of a process plant). The preliminary study [Kout *et al.*, 1997] proved that the set of the on-line measured attributes of motor currents offers a good potential for industrial diagnostics of pump defects when combined with AI methods, namely with machine learning (ML) techniques. The goal of the accomplished experiments was to develop a methodology allowing to classify any (unseen) case characterised by these on-line measured attributes into one of the five classes indicated in the training data by the attribute STATUS. The attribute STATUS is considered to be a dependent attribute reaching the following states: **0** indicates normal state, **1** indicates low-level cavitation, **2** cavitation, **3** low-level blockage and **4** stands for blockage.

This paper describes a fault diagnostic system based on *competence estimate* (or fuzzy) definition of the classified parameter. ML techniques are applied on the training set of examples using the new description of the classified parameter and finally they are used in order to transform classifications of unseen cases back to a crisp form. In case of the intelligent pump diagnostics, the crisp parameter STATUS is substituted by three competence estimate (membership) functions from the universe of attribute space.

2 Method description

Fault diagnostic methods are requested to generate unambiguous results – they have to discriminate normal cases from the fault ones. Moreover a type of the fault is usually to be determined besides. The methods are supposed to give the response in the form of a clear single valued classification as hazy responses can be hardly used for control of an observed system. This requirement implies that unambiguous classifications have to be assigned to all the training examples used during the learning period as well. However, this type of assignment does not correspond to reality when smooth continuous transitions connect adjacent classes. From this point of view, it seems to be more natural to allow for fuzzy representation of class competence.

The method being described can be divided to four fundamental steps (explained in the rest of the paper):

1. Extension of the set of considered attributes by estimates of competence to individual classifications— each possible classification *i* is estimated by a *competence estimate function* CEF_i; the set of values of all considered CEFs for any example defines its *competence estimate vector CEV*,
2. search for relation between CEVs and the single valued classifications -- proposal of a method how to find the single valued classification from CEFs (creation of a corresponding decision tree or a neural net),
3. assessment of CEVs for the testing examples (using CEVs in the training set and N nearest neighbors method),
4. transformation of a competence class estimate description of testing examples to a single value.

Process denoted in the step *1* depends mostly on real application. The following chapter suggests a process leading to the above-mentioned extension. The proposal can be easily adapted to real application specification if the individual classes are totally or partially ordered. As the extension result, there is generated a set of training examples enriched by competence estimate functions CEFs. Assessment of CEFs for the testing examples is achieved by N-nearest neighbors algorithm. The database of training examples is searched through. The nearest training examples are sorted out for each testing example. Then, the competence estimate design of each tested example is proposed from the weighted classification competence estimates of the selected nearest neighbors. Finally, there are applied in parallel results of two ML techniques, namely ID3 and NN algorithms, to transform classification estimates or competence estimate vectors of tested examples to single valued classifications - (see Figure 1).

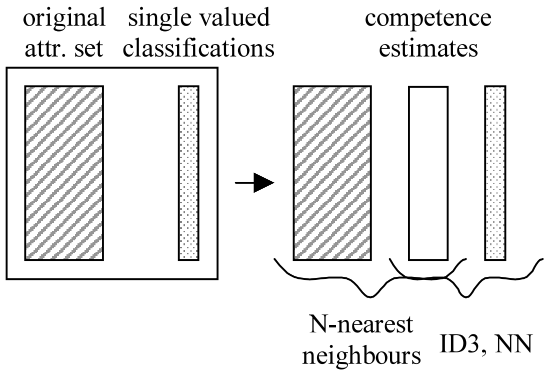

Figure 1: Competence Estimate Method

It is worth to notice that steps 1, 2 and 4 are conformable to fuzzification and defuzzification steps in fuzzy classifier systems. However, by the same mail it is obvious that the suggested classification method does not satisfy definition of fuzzy classifier system: *"A fuzzy classifier system is a system which learns fuzzy rules in order to guide its performance in an arbitrary -environment"* [Geyer-Schultz, 1995]. Competence estimates append utilisation of background knowledge that can not be extracted from the attribute set nor the single valued classifications.

To sum it up, the suggested method is suitable for all tasks that meet the following requirements:

i. Training and testing examples are described by means of a set of attributes,
ii. there is assigned a single class to each training example (by a teacher/domain expert),
iii. the set of individual classes is totally or partially ordered (again with aid of background knowledge provided by the domain expert),
iv. transitions between adjacent diagnostic states show continuous course, while original classes corresponding to these states change discretely (very representative example of such a case is explained in case study - see Section 3),
v. competence estimate functions transforming the universe of attribute space to the real unit interval<0,1> can be found.

2.1 Competence Estimate Definition of Classified Parameter

Let's consider an application with **n** different single valued classifications bound by a relation of partial order. Any longest ordered sequence of classifications is referred to as an **ordering**. Suppose there are **m** different orderings in the considered set of classifications. Generally, these orderings do not have to be necessarily mutually disjoint and each classification should appertain to one ordering at least. In the case of fault diagnostics, it usually holds that each ordering $\mathbf{o_i}$ corresponds to a transition from a normal state **norm** to a marginal fault state $\mathbf{f\text{-}fault_i}$ (final fault). Then, an individual CEF $\mathbf{f_i}$ qualifies a competence of a selected case to the i-th fault scale. The value P_i^A of the CEF f_i evaluates competence of the example A to the i-th marginal classification $\mathbf{f\text{-}fault_i}$ ($f_i(A)= P_i^A$). Provided P_i^A is equal to or near to 0, the example A shows to be normal (from the point of view of this type of fault). Provided P_i^A is equal to or near to 1, the example A matches with the state $\mathbf{f\text{-}fault_i}$. Otherwise, the example A is going to be classified as a state from the i-th fault scale: belief in this classification corresponds to P_i^A.

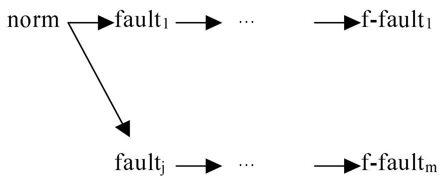

Figure 2 Fault state diagram

Having the **m** competence estimate functions corresponding to faults, it is natural to introduce the last competence estimate function for the normal state. It is denoted by P^A_{m+1} and it symbolically complements the sum of all probabilities to 1. Therefore, the following equation holds for all examples:

$$\sum_{i=1}^{m} P_i^A = 1 - P_{m+1}^A$$

where m is number of particular competence estimate functions,
P_i^A is a value of the i-th particular competence estimate function of example A.

The crucial phase of the whole method comes with definition of competence estimate functions f_i, which have to be based on some form of background knowledge available in the given domain. There are basically two extreme situations when the method can not be applied. The first situation appears when these functions can not be formulated at all (not even intuitively). The second extreme occurs under condition that competence estimate functions can be defined absolutely precisely, i.e. for each $i \leq m$ there exists function φ_i such that:

$$f_i(A) = \varphi_i\left(a_1^A, \ldots, a_k^A\right)$$

where f_i is the i-th particular competence estimate function,
a_i^A is a value of the i-th attribute of example A,
k is the number of measured attributes.

Of course, the method does not fail under such circumstances, but it makes no use to apply it having another perfect tool for classification. In such a case, all the values of competence estimate functions could be analytically calculated for unseen examples and if they were perfect they would be easily transformed to a single value. On the other hand, the method can be applied in all other cases. Especially if class competence estimate design of the training examples is intuitive or it can be mathematically defined, but attributes do not occur as the input parameters of the function (individual functions can be defined with respect to the selected attributes, which are not available within the testing set).

2.2 N-Nearest Neighbours

The core of classification method of N-nearest neighbours is very simple provided vague terms neighbour, similarity etc. are taken in account. In fact, these terms have to be defined through relatively complicated set of numeric parameters, whose adjustment represents the most difficult and time-consuming procedure within the framework of method of this type.

Let us define these vague terms more precisely. **The nearest neighbour** (let us denote it B) to a testing example A is defined as the example with the shortest weighted distance of its image from the image of A in the normalised attribute space of dimension ***k***:

$$D = \sum_{i=1}^{k} w_i * \frac{\left|a_i^A - a_i^B\right|}{\max\left(a_i\right) - \min\left(a_i\right)}$$

where k is the number of measured attributes,
a_i^A is a value of attribute a_i for the example A,
a_i^B is a value of attribute a_i for the case B,
$\max(a_i)$ is the maximum value of the attribute a_i in the training set,
$\min(a_i)$ is the minimum value of the attribute a_i in the training set,
w_i is the weight associated to the attribute a_i.

The value of the competence estimate function P_j within each considered scale of the example A is again

$$P_{jA} = \frac{\sum_{i=1}^{N} v_i * P_{ji}}{\sum_{i=1}^{N} v_i}$$

calculated as a weighted sum of corresponding competence estimate functions of N nearest neighbours of the example A:

where N is the number of the nearest neighbours used for prediction,
v_i is a weight associated with the i-th nearest neighbour of example A,
P_{ji} is a value of the j-th competence estimate function of the i-th nearest neighbour of the example A.

It follows that in order to reach good results of prediction two different arrays of weights (***w, v***) had to be optimised together with the parameter **N**.

2.3 ID3 – Return to Original Single Valued Classification

The ID3 algorithm, member of TDIDT (Top-Down Induction of Decision Trees) family, is a well-known algorithm of creating decision trees from examples [Quinlan, 1993].

The TDIDT algorithms do not use any form of background knowledge. The inference engine constructs a decision tree in a recursive way from its root to its leaves, partitioning the original training set into smaller and smaller subsets. Inducing a decision tree exploits a general strategic heuristic: simpler trees are preferable because they exhibit better predictive power. There exist several pre- and post- pruning techniques exploited by TDIDT algorithms for processing noisy data and reducing the “bushy” decision trees [Quinlan, 1993].

The idea of transformation of competence estimate classification values to the single valued design is quite easy. A subset of training examples with known original classification is employed for this purpose. The competence estimate classifications are calculated on all the training examples belonging to the selected subset using the method described in section 2.2. In such a way, a new set of training data is introduced. Each object is represented by attributes corresponding to competence estimate classifications, desired class agrees with the single value original classification. A decision tree can be generated for the obtained data set. This tree gives a chance to return to the single value classifications for the testing examples after N-nearest neighbour method application.

And there is one more advantage of this approach. ID3 algorithm attempts to reach the optimal depth of the generated tree [Russel and Norvig, 1995]. A tree, which answers basic conditions of probably approximately correct (PAC) learning, must meet the following inequality:

$$\ln ex\_num \geq \ln at\_num^{Hl}$$

where ex_num is size of the training set,
at_num is number of attributes,
Hl is the optimal depth of the tree.

Usually it holds that **m<k** (number of considered class orderings is significantly lower than the number of measured attributes). Thus, ID3 does not need as many training examples to generate a tree of the same depth ensuring the same degree of reliability as in the case of direct learning on the single value classifications.

2.4 NN – Return to Original Single Valued Classification

The neural net architecture used for this study is a feedforward network with one hidden layer. We have used the standard backpropagation-training scheme that employs the delta rule-training algorithm. A detailed description of the neural net architecture and the delta rule training algorithm used can be found in [Soucek, 1991, Pao, 1989].

This neural net scheme is employed using supervised learning. Upon reaching a minimum error threshold or improvement level the neural net is considered trained and the weights and other parameters previously adjusted are clamped or fixed. The trained neural net is then used to classify unknown samples or in our case, characterise pump operation from sampled input parameters. This is performed by presenting characteristics from an unknown operating state to the neural net that is processed in a feedforward manner and an output value is computed. The output value represents the computed state of the system.

The most appropriate network topology is dependent on the training data set and the mapping surface to be encoded in the neural net. There is no general theory how to construct the best topology for a given problem although general rules of thumb do exist. In classification tasks such as this one (linearly non-separable), three-layered networks are used often. The number of input nodes corresponds to the dimension of the input vector. In the case of pure NN approach, there are 8 input neurons with each one representing each input attribute. In the case of the combined method (combination of competence estimate and neural network approaches), there were employed 3 neurons in the input layer, one for each competence estimate value. The number of output nodes may be a single node or multiple nodes with each node representing one of the possible attributes or classification values. For our study we use 5 output neurons with each one representing one of the possible operating states of the system. The data and the states of the system are described in the following section. The number of nodes in a hidden layer is usually determined experimentally. There are several heuristics, however they are of a general nature and merely provide a good starting point for defining the initial neural net architecture. Subsequent training and testing sessions are typically needed to refine the neural net structure and improve classification accuracy.

2.5 Recapitulation

Having defined individual method steps in detail, it is worth to summarise exact sequence of these steps at full length. Compliance of the sequence assures correct segregation of training and testing data.

Let us have a training set ***T*** consisting of ***t*** examples. Each example is assigned to one of ***n*** different single valued classifications, which belong to ***m*** different orderings. The goal is to assign correctly one single valued classification to each of ***u*** patterns belonging to a testing set ***U***. The process of classification is as follows (steps 1 to 3 represent learning phase, steps 4 to 6 testing phase):

1. Define CEF $f_1,\dots,f_m$ and provide their values for all examples from ***T.*** Additional competence estimate for the normal state is defined as the complement of $(f_1+\dots+f_m)$ to 1. Thus m+1 new values are assigned to each example $(P_{(i,1)},\dots,P_{(i,m+1)})$,
2. The N-nearest neighbour algorithm is applied to the set ***T*** to generate new transformed values CEVT=$(P^T_{(i,1)},\dots,P^T_{(i,m+1)})$ for each example belonging to ***T.*** The parameters v for the N-nearest neighbour algorithm are adjusted so that the total difference between CEV and CEV^T is minimised on ***T.*** The CEV^T values will be used in the next steps to ensure maximal similarity between the training set and the testing set data.
3. Create a decision tree (learn a neural net) that transforms values obtained in the step 2 back to single valued classifications; all patterns from set ***T*** are used for this purpose; values $(P^T_{(i,1)},\dots,P^T_{(i,m+1)})$ are considered as attributes, single valued classifications represent desired classes.
4. Assess values of CED for each example belonging to the set ***U*** using N-nearest neighbour algorithm– values $(P^T_{(i,1)},\dots,P^T_{(i,m+1)})$ are assigned to each testing example.
5. Use the decision tree (neural net) created in the step 3 to obtain single valued classifications for all examples from ***U***.
6. Evaluate quality of the generated classifications (provided the correct classifications are known within the set ***U***).

3 Case Study

Pumps, motors and bearings constitute the major share of equipment, which require very frequent maintenance in the process industries. The unscheduled shutdowns, which they cause, result in considerable production and revenue loss. Incipient fault detection based on advanced condition monitoring techniques would ensure lower down time and greater throughput. Condition monitoring of pumps has not received much attention and there are only few papers in which schemes are proposed for the fault detection of pumps by analysing the signal derived from a pressure transmitter or a flow-meter installed in the pumping system. However, it is possible to detect some fault conditions in pumps by analysing the stator current of the motor driving the pump.

Number of parameters can be derived from the measurements on the system consisting of a motor and a pump. Throughout this study there was used a set of nine parameters. Eight parameters represent significant magnitudes of the stator current spectrum. The last parameter, denoted by STATUS, provides the classification of the actual behaviour exhibited by the pump system - its diagnoses. There can be distinguished 5 important states, namely **normal** function and four fault types. As a matter of fact, these five states can be partially ordered as there are just two marginal faults associated with the observed pumps: **cavitation** and **blockage**. Thus there can be defined two different state orderings, each of them corresponding to one of marginal faults: ordering A – normal state (denoted 0), low-level cavitation (1) and cavitation (2) and ordering B – normal state (0), low-level blockage (3) and blockage (4).

As it follows from the method description chapter, two competence estimate functions evaluating competence of each example to one of the fault orderings should be defined. In addition to that, the third CEF expressing competence of example to a normal state should be calculated (it makes sum of all CEFs for a given example equal to 1). Despite of the fact the functions can be hardly analytically qualified on the above-defined attribute set, they can be related to parameters from outside of the attribute set. All the measurements within the training set were collected in such a way, that faults were simulated by gradually closing the valve. The cavitation was simulated with aid of valve modifications in pump inlet, the blockage was caused by gradually closing outlet valve. Consequently, definition of competence estimate functions can be based on position of the valves – the more is inlet valve closed the higher is measurement competence to cavitation (by analogy for blockage). Of course, positions of valves do not define the state absolutely clearly (their position is linearly changed while pump conditions do not change linearly). One more criterion can be used to make desired competencies more precise. A water flow rate was measured during the pump experiments simultaneously with stator current measurements. The flow rate measurement is competitive condition monitoring technique and can not be used as an extra attribute during learning phase. Nevertheless, it can help to set down competence estimate classification of the individual training examples.

3.1 Reached Results And Their Comparison To Results Of Traditional ML Algorithms

There was measured a centrifugal pump with magnetic coupling during experiments. The results achieved by the suggested competence estimate method are summarised in Tables 1 and 2. They are compared with results achieved by two other well-known ML techniques – neural nets and ID3. Both methods dealt with single valued classifications. Table 1 compares results reached by pure ID3 method with results of competence estimate method combined with ID3 for return to the crisp valued classifications. Likewise, Table 2 shows difference between correctness of NN method and competence estimate method using NN to return to single value classifications. There are two different types of classification errors defined. The first type simply involves *all misclassifications*. Error occurs whenever the desired class does not agree with the generated classification. The second type pays attention to such called *hard errors only*. These hard errors do not occur whenever the suggested classification C_1 is not equal to the original one C_0, but C_1 is direct predecessor or successor of C_0. It means, that class 0 can be classified as 1 or 3, the class 1 can be classified as 2 and class 3 as 4 (all the statements hold vice versa too). A hard error is not considered when it is correctly assigned a fault scale and classification is directly adjoined to the desired class.

DistrNo/ Method	ID3		Comp. estimate method with ID3	
	all [%]	hard [%]	all [%]	hard [%]
Distr 1	85	94	90	100
Distr 2	74	94	85	98
Distr 3	76	96	84	98
Average	78.3	94.7	86.3	98.7

Table 1 Correctness of pump fault classification – ID3 approach (**all** denotes correctness when all errors are considered, **hard** denotes correctness when only hard errors are considered)

DistrNo/ Method	NN		Comp. estimate method with NN	
	all [%]	hard [%]	all [%]	hard [%]
Distr 1	92	100	91	100
Distr 2	88	98	91	98
Distr 3	86	99	88	99
Average	88.7	99	90	99

Table 2 Correctness of pump fault classification – NN approach

There was used a set of 422 examples at all. The competence estimate classifications based on input and output valve positions and water flow rate were established for all the measured examples. Then, there were learned new competence estimate classifications for all the examples (they were used for training and testing at the same time). It was done in 422 steps, one of the examples was always chosen, 6 nearest neighbours were found within the rest of 421 examples and a new competence estimate classification was assigned to it. As far as the number of the nearest neighbours is taken into consideration, 6 neighbours showed to be optimum number for given amount of examples. The weight array **w** was set in conformity with correlation between individual attributes and the desired class (w=[2, 1.6, 1.4, 1.2, 1, 0.8, 0.8, 0.8]). Finally, decision trees had to be generated and neural nets learned. Three different random distributions were used for this purpose, since the examples can not be used for training and testing at the same time when the decision trees are generated. The training set consisted always of 322 examples and testing set included the remaining 100 examples. The relation between the desired classes and classifications is shown in Table 3 (600 testing examples – 3 distributions with 100 examples, 2 approaches). The hard errors are marked out in bold.

Class/	0	1	2	3	4	Avg

Classi-fication						corr. [%]
0	236	9	0	5	0	94.4
1	13	15	0	0	0	53.6
2	**2**	1	129	0	0	97.7
3	7	0	**3**	41	12	65.1
4	0	0	**2**	16	109	84.6

Table 3 Confusion matrix for competence estimate method with ID3 and NN

3.2 Method Modification

There exist many possible changes of dealing with estimates of class competence indeed. One more method modification was practically implemented. In the first instance, three etalons of marginal classifications (normal state, cavitation, and blockage) were determined. They were computed as average values of attributes taken from measurements with extreme position of valves. As soon as there are etalons of the marginal classes established, competence of individual examples to classes can be expressed in accordance with their distances to these etalons.

Having distances d0, d1 and d2, values of competence estimate functions are calculated as relative similarity measures to introduced etalons. Similarity measure is introduced as an inverse of distance from the etalon:

$$P_i^A = \frac{\frac{1}{d_i^A}}{\sum_{k=0}^{2} \frac{1}{d_k^A}}, i = 0,1,2$$

The closer the example image to a class etalon in attribute space is the higher value of the appropriate probabilistic function. At the same time it holds:

$$\sum_{k=0}^{2} P_k^A = 1$$

Transformation of competence estimate classification values to the single valued design is done again by ID3. Experimental results achieved by this approach are shown in Table 4.

DistrNo/ Method	Modified method	
	all [%]	hard [%]
Distr 1	84	98
Distr 2	80	97
Distr 3	80	98
Average	81.3	97.7

Table 4 Correctness of modified competence estimate method

This modified method does not reach as high absolute correctness of classification as the method using competence estimate functions, but it still shows better results than ID3 and NN methods applied directly to the attribute set. In comparison to the competence estimate method, it does not demand such profound background knowledge. There just have to be known enough cases representing all marginal fault states and the normal state. Consequently, the modified method can be applied whenever it is too difficult to design commensurate competence functions.

4 Conclusion

In this paper, there has been proposed a new competence estimate approach to process fault diagnosis. The method is based on well-known machine learning techniques, the originality of the approach is in their combination in conjunction with the competence estimate classification. The basic source of the competence estimates for the training set of examples is background knowledge available due to the setting of the experimental environment. This does not have to be true in general, namely in the real conditions. That is why, there has been suggested a modification of presented method which enables processing of tasks that are hard to be decomposed according to class competence estimate method scenario.

The method was applied on monitoring process results in order to detect possible fault states. The results obtained indicate that it is possible to diagnose certain pump problems without any additional instrumentation, i.e. the set of the on-line measured attributes offers a good potential for industrial diagnostics of pump defects when combined with suggested method. Moreover, a comparison of correctness of classifications proves that the suggested method shows better results than direct application of traditional ML techniques.

Acknowledgements

This research was partially supported by the grant CEZ J04/98:212300013.

References

[Geyer-Schultz, 1995] Geyer-Schultz, A.: Fuzzy Rule-Based Expert Systems and Genetic Machine Learning, Physica-Verlag Heidelberg, Germany, 1995

[Kout *et al.*, 1997] Kout, J., Klema, J., Lauryn, Š., Štěpánková, O., Mařík, V.: AI Methods for Intelligent Pump Diagnostics II., Research Report GL-27/97, CTU Prague

[Quinlan, 1993] Quinlan, J., R.: C 4.5, Programs for Machine Learning, Morgan Kaufmann Publishers, California, 1993

[Russell and Norvig, 1995] Russell, S., Norvig, P.: Artificial Intelligence, a modern approach. Prentice - Hall, 1995

[Soucek, 1991] Souček, B. and The IRIS Group: Neural and Intelligent System Integration, A Wiley-Interscience Publication, New York - Chichester - Brisbane - Toronto - Singapore, 1991

[Pao, 1989] Pao, Yoh-Han: Adaptive Pattern Recognition and Neural Networks, Case Western Reserve University, Addison - Wesley Publishing, 1989