A NOVEL FAULT DIAGNOSIS SYSTEM FOR AIRCRAFT BASED ON ADABOOST AND FIVE SUBSYSTEMS WITH DIFFERENT PATTERN RECOGNITION METHODS

ZE-FENG WANG[1], JEAN-LUC ZARADER[2], SYLVAIN ARGENTIERI[3]

[1 2 3]Institute for Intelligent Systems and Robotics (ISIR)
University Pierre and Marie Curie (UPMC) - Sorbonnes Universities, Paris 75005, France
E-MIAL: zwang@isir.upmc.frn, zarader@isir.upmc.fr, argentieri@isir.upmc.fr

Abstract:

The goal of this paper is to devise a fault diagnosis and decision system of aircraft and practice it in real flight condition. In order to face the real usage of system and keep its reliability and safety, the system contains two monitoring terminals according to different requirements and equipment conditions. One is on operation at the aircraft (on-line), which is based on ADABOOST with ten weak classifiers. It only needs to detect if there is any fault or not, and make two judgments of the health of aircraft - normal or mild dangerous (- fault tolerant system can resolve it) and dangerous (- back to airport) for pilot. Another one is for maintenance office (off-line), which is based on five subsystem with different recognition methods: Back-Propagation Neural Networks (BP), Probabilistic Neural Networks (PNN), Learning Vector Quantization Neural Networks (LVQ), Gaussian Mixture Models (GMM) and Decision Tree (DT). With the fusion of the diagnosis results of these five subsystems, the system can detail the diagnosis results and distinguish each fault.

Keywords:

Adaboost, Neural Networks, BP, PNN, LVQ, GMM, DT

1. Introduction

With the development of aerotechnics and requirements of aircraft performance, the system of aircraft becomes multi-functional and more complex [1]. Such a complex system, it is difficult to fault diagnosis by traditional methods, which are performed by expertise, by observing how the system works (generally through unit tests on each component of the complex system) [2]. It costs more time and more human resources. Through modern aircrafts generally have a good robust performance and the fault tolerant system can neutralize some faults, it arouses more hidden danger. The small symptoms of faults are covered and it becomes more difficult to detect them by maintenance staff. If they can not be found in time, maybe it leads to some terrible accidents. Consequently, aircraft needs to embed a fault diagnosis system to achieve self-diagnosis and the system needs to be intelligent to analysis different condition of aircraft, give an alarm to pilot in necessary and send a report to engineer and maintenance staff.

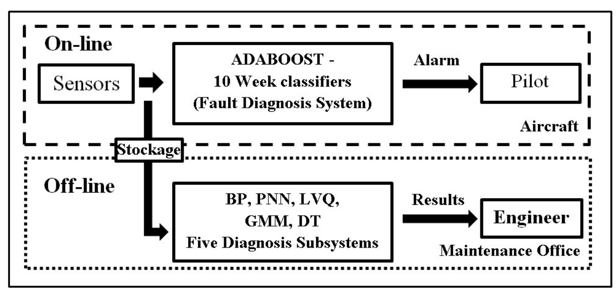

Figure 1. Schema of Faults Diagnosis System

In this paper, a novel fault diagnosis system for aircraft is built. It can detect the small symptoms of faults and diagnosis them. For the sake of practical applicability and keep reliability and safety of system, two different monitoring terminals are designed, which are according to the different situations requirement and equipment conditions. One is on operation at the aircraft (on-line), based on Adaboost with ten weak classifiers. It needs only to detect fault and make two judgments of the health of aircraft - normal or fault for pilot. It doesn't need expensive or special advanced equipments to do the calculations and it can update the machine learning in a short time by ADABOOST [3]. This is very useful in the some special or abnormal conditions.

For example, if fight aircraft lose some sensors during its mission and there is no time or no source to repair them in time, the system have to build a temporary fault diagnosis system with the rest sensors.

Another monitoring terminal is for the maintenance office (off-line), which is based on five different recognition methods: BP, PNN, LVQ, GMM and DT. With the fusion of the diagnosis results of these five subsystems, the system can detail the diagnosis results and distinguish each fault. In comparison with some other methods, this system is more convenient and suitable for using. It doesn't have the problem of conflict of new fault rules, which leads a big problem in expert system. On the basis of complementary advantages, the five different subsystems improve the pragmatic cooperation and come to a convincing conclusion of diagnosis.

The rest of this paper is organized as follows; Section II demonstrates the monitoring terminal on-line based on Adaboost with its ten weak classifiers; Section III reveals the structure of the monitoring terminal off-line with the five subsystems. In Section IV, the system is tested by an aircraft Y which is from Dassault Aviation, the results are presented and analyzed in this section; and the last Section concludes the paper and presents a preview of future works.

2. Monitoring terminal on-line based on ADABOOST

In modern aircraft, not all the faults are evident. There are many subtle faults which are difficult to perceive, but dangerous to aircraft and pilot. For example faults of aviation oxygen supply system, faults of aviation anti-icing system, some hidden faults of engine. Thus, the monitoring terminal on-line is indispensable which keeps watch on these aviation systems, send warning to pilot if there is any fault. For this on-line case, the system doesn't need to delves into the faults diagnosis, because there is no time to waste. Adaptive Boosting (Adaboost) is a machine learning algorithm, which can be used in conjunction with many other learning algorithms to improve their performance. It is a very successful technique for solving the two-class classification problem. It consists of weak classifiers and the weight of each weak classifier is trained during the learning to build a strong classifier. For each call of weak classifier, a distribution of weights is updated that indicates the importance of examples in the data set for the classification [4]. On each iteration, the weights of each incorrectly classified example are increased, and the weights of each correctly classified example are decreased, so the new classifier focuses on the examples which have so far eluded correct classification.

Given a training set $\{(x_i, y_i)\}$
where $i = 1, ..., N$, $x_i \in X$ and $y_i \in \{-1, +1\}$
Initialize $t = 1$, example distrib. $D_t(i) = 1/n \; \forall i$ and $E = 1$
while $E > \xi$ **do**
  Train weak classifiers using D_t
  Get weak hypothesis $h_t : X \to \{-1, +1\}$
  Compute weak class. weighted er. $\epsilon_t = \sum_{h_t(x_i) \neq y_i} D_t(i)$
  Choose $\alpha_t = \frac{1}{2} \ln(\frac{1-\epsilon_t}{\epsilon_t})$
  Update $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{\sum_i D_{t+1}(i)}$
  Compute strong hypotheses $H(x_i) = sign(\sum_t \alpha_t h_t(x_i))$
  Compute strong classifier error $E = \sum_{H(x_i) \neq y_i} \frac{1}{N}$
  $t++$
Output the final hypothesis $H(x) = sign(\sum_t \alpha_t h_t(x))$

Figure 2. Algorithm of Adaboost

As shown in the Figure 1, the sensors information are treated as the inputs of system (x_i), and the outputs (y_i) is either +1(fault) or -1(no fault). D_i is the weight of each weak classifier, which is trained by α_t via the error ϵ_t in each iteration [5].

In this paper, the neural network Back Propagation with training method Levenberg-Marquardt (BPLM) is chosen as the weak classifier. It is a very simple, but robust, method for classification. Neural networks can be viewed as highly nonlinear functions with the basic the form as Equation 1. x are the inputs, y are the outputs and w are the weights of network. The update of w is achieved by δ, which can be found by Equation 2, where J is the jacobian matrix for the system, λ is the Levenberg's damping factor, δ is the weight update vector that the training system wants to find and E is the error vector containing the output errors for each input vector used on training the network [6]. The λ damping factor is adjusted at each iteration and guides the optimization process.

$$F(x, w) = y \tag{1}$$

$$(J^t J + \lambda I)\delta = J^t E \tag{2}$$

In comparison with other training methods, like Gradient descent with momentum and Quasi-Newton, the training by Levenberg-Marquardt (LM) is more fast, correct and robust. In addition to this, LM is sensitive to initial weights of networks, this is another reason to chose this network as weak classifier. According to different initial weights, there is difference between every two weak classifiers. Thus, on the same database, each weak classifier can obtain a different result. Adaboost combines these weak classifiers as a strong classifier and reaches a conclusion.

3 Monitoring terminal off-line based on five different recognition methods

The monitoring terminal off-line works for the maintenance office. It analyzes all the sensors information which are stored during the flight mission and diagnoses clearly every fault. In order to ensure the reliability of fault diagnosis system, this monitoring terminal consists of five sub-systems based on five different recognition methods (BPLM, PNN, LVQ, GMM and DT). At last, the system will combine these five diagnosis results and reach a conclusion.

3.1 Subsystem based on PNN

Different from BP, PNN is a probabilistic neural network that can compute nonlinear decision boundaries which approach the Bayes optimal is formed [7]. The theory can be concluded as: if the probability density function (pdf) of each of the populations is known, then an unknown, $x = (x_1, x_2, ..., x_n)$ a vector with n attributes, belongs to class i if:

$$f_i(x) > f_j(x), i \neq j \tag{3}$$

where f_i and f_j are the pdf for class i and j. Then the classification decision becomes:

$$h_i \cdot c_i \cdot f_i(x) > h_j \cdot c_j \cdot f_j(x), i \neq j \tag{4}$$

where h is the Prior probability (probability of an unknown sample being drawn from a particular population) and c is the Misclassification cost (cost of incorrectly classifying an unknown).

In general, PNN has four layers: Input layer, Pattern layer, Summation layer and Output layer. In the input layer, each neuron represents an sensor of aircraft. When a input vector is present, pattern layer computes the distance from the input vector to the training input vectors. It produce a vector where its elements indicate how close the input is to training input. The summation layer sums the contribution for each class of inputs and each class has only one cell in this layer. Finally, a compete transfer function on the output picks the maximum of these probabilities, and produces a "1" for that class and a "0" for the other classes [8]. As a whole, the PNN can be presented as Equation 5, where x_{ik} is "k^{th}" training pattern from category i, δ is a smoothing parameter, m is the total number of training patterns and p is the length of vector.

$$f_i(x) = 1/((2\pi)^{p/2}\delta^p m)\Sigma exp[-(x - x_{ik})^2/(2\delta^2)] \tag{5}$$

3.2 Subsystem based on LVQ

LVQ algorithms do not approximate density functions of class samples like Probabilistic Neural Networks do, but directly define class boundaries based on prototypes, a nearest-neighbor rule and a winner-takes-it-all paradigm. Thus LVQ neural network is a competitive net with supervisors. It has only one hidden layer of neurons, fully connected with the input layer. This hidden layer is a competitive layer, which learns to classify input vectors in much the same way as the competitive layers of Self-Organizing Feature Maps. The linear layer transforms the competitive layer's classes into target classifications defined by defined by the types of faults. The net calculates the distances between the inputs and each neuron in the competitive layer as shown in Equation 6, where w_{ij} is the weight from input neuron i and hidden neuron j, x is the inputs. Finally, the nearest neuron is the winner who send '1' to a neuron in the output layer according its target class [9].

$$d_i = \sqrt{\sum_{i=1}^{n} (x_i - w_{ij})^2} \tag{6}$$

Moreover, LVQ algorithm of neural network can be improved which is named LVQ2. In this case, there is a region of winner, rather than only one neuron of winner. That is to say, the input falls into a "window" near the midplane of the two vectors, which window is defined as:

$$min\{\frac{d_i}{d_j}, \frac{d_j}{d_i}\} > \rho \tag{7}$$

where ρ is the width of window, generally it is taken as $2/3$. With the supervisor of target classes during the machine learning, the weights of networks is adjusted by Equation 8 and 9, where η is the speed of learning. If the classification is correct, the weight w is update as Equation 8; if there is a misclassification, w is update as Equation 9.

$$w_{ij} = w_{ij} + \eta(x_i - w_{ij}) \tag{8}$$

$$w_{ij} = w_{ij} - \eta(x_i - w_{ij}) \tag{9}$$

3.3. Subsystem based on Gaussian mixture models

GMM is a parametric probability density function represented as a weighted sum of Gaussian component densities. Its parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm or Maximum a posteriori (MAP) estimation from a well-trained prior model [10]. In the Gaussian case, a GMM is a simple linear superposition of Gaussian components, which aims at providing a richer class of density models than a single Gaussian. For a model of M Gaussian states, a GMM density function of a variable X_n can be defined as:

$$p(X|\lambda) = \Sigma_{i=1}^{M} p_i b_i(x_n) \quad (10)$$

where p_i is the probability of being in the state i and b_i is the Gaussian density function. The set of parameter λ can be written as $\lambda = \{p_i, \mu_i, \Sigma_i\}, i = (1, ..., M)$, which represents the set of weights p_i , mean vectors μ_i and covariance matrices Σ_i of the GMM.

In the faults diagnosis task, an state GMM is associated to each aircraft situation(S) including the normal condition and each type of faults. On the basis, the aim is to determine which model of situation $\widehat{S}$ has the biggest a posteriori probability over a set $X = x_1, x_2, \ldots, x_n$, which are from the measurement of sensors. So, according to Bayes rules, $\widehat{S}$ is:

$$\widehat{S} = Argmaxp(\lambda_k|X) = Argmax\frac{p(X|\lambda_k)p(\lambda_k)}{p(X)} \quad (11)$$

where, λ_k represents the mixture parameterizations of the M-state GMM associated to the k^{th} situation [11] of aircraft. So in this subsystem, the goal of machine learning is to determine the the parameters λ_k for each situation of aircraft. The system use the classical iterative Expectation - Maximization (EM) to determine them.

3.4. Subsystem based on Decision Tree

Decision tree learning is a method commonly used in data mining. Its goal is to create a model with a hierarchical tree structure that predicts a target value, based on several input values that describe the example. It checks input variables and makes corresponding decisions like classification. Like the system of expert, learned trees can also be re-represented as sets of if-then rules to improve human readability, but they don't need human experts to establish a knowledge and rules database, they build rules by themselves. This algorithm uses entropy to determine the structure of tree: the root and branches. Entropy characterizes the impurity of an arbitrary collection of examples. For example, given a collection S containing positive and negative examples of a target concept, the entropy of S relative to this Boolean classification is:

$$Entropy(S) = -P_{\oplus} log_2 P_{\oplus} - P_{\ominus} log_2 P_{\ominus} \quad (12)$$

where $P_{\oplus}$ and $P_{\ominus}$ are respectively the proportions of positive and negative examples in S ($0 * log(0)$ is considered as 0 here). In case here, a collection will be the set of values provided by sensors. The system calculates the entropies of all the variables. Then, a measurement of the effectiveness of an attribute or value, which is called information gain, is calculated by entropy to determine root attribute and the branches attributes. As Equation 13 shown, the information gain, Gain(S,A), of an attribute A, relative to a collection of examples S, is defined as:

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in (A)} \frac{S_v}{S} Entropy(S_v) \quad (13)$$

where values(A) is the set of all possible values for the variable A, and S_v is the subset of S for which the variable A has the value v. The variable with the highest information gain is the root node from which leaves go out, each leaf corresponds to a value that its variable can take. Following this, entropies and information gains corresponding to the other variables and to each of the leaves going out of the root nodes are computed, and new nodes are taken corresponding to the highest gains. And so on, until all variables are used and the final tree is formed. Classification Tree is done by using the values of the variables constituting an example and following the resulting tree leaves and nodes from the root to the final leaf which gives the class. In addition to this, all the input variables in the system are not Boolean but multiple values. Therefore, the values of each attribute are sequently arrayed firstly, and then all the expected pair values which are neighboring but sorted in different classes are picked out. After that, all possible threshold values are calculated, as the halves of the sums of the pairs mentioned before. Finally, the threshold value with maximum information gain is taken.

4 RESULTS OF EXPERIMENTS OF AIRCRAFT

4.1 Experiment Drawing

The Fault diagnosis system in this paper is tested on an engine system of aircraft Y, which is from Dassault Aviation. This engine system is a complex system, it not only consists of engine(s), but also contains a complex tube system and some subsystems, like anti-icing system, which uses the hot air from engines to warm wings and fuselage. There are 33 sensors installed on the aircraft, which work for this fault diagnosis system. They are about temperature, pressure, debit of air, system variables (for example to indicate the position of the valve), speed and altitude of aircraft. The measurement of these sensors constitutes a database consisting of 320000 examples (320000 records of flight, one record per second), they are used to build the fault diagnosis system and test it. Data is measured during a flight and describes the faults and the normal state without faults. 75% of the data are chosen randomly for the training and the remaining 25% are used to test the system.

4.2 Results of Test on the monitoring terminal on-line

On the monitoring terminal, the Adaboost system consists of ten weak classifiers based on BPLM. As shown in Table 1, although these classifiers based on the same algorithm, they have differences from each other. It confirms that as described by Section II. The advantage of Adaboost is clearly shown in this table, the strong classifier obtain a much better result, its correct detection of fault is 99%.

But, there still exists a big problem: all the data is recorded per second during the flight mission of the aircraft, so even if there is only 1% missed detection ratio as shown in Table 1, there will be 1 time of miss detection per 1.67 minute, and 4.29% false alarm means that the pilot will receive 3 times of false alarms per minute. These conditions are unacceptable in the real applications. Thus, in order to solve this problem, it should consider a fusion of the fault diagnosis results of a period time, as 15 or 30 seconds. That is to say, the system records the fault diagnosis results from a time; then compares the faults ratio diagnosed in the results with pre-set threshold; at last judges the situation of aircraft. For example, the pre-set threshold is 20%. Since the frequency is below the threshold, the diagnosis of these 30 seconds is considered as no-fault. This criterion is used to test the continuous data of 7800 seconds (a flight mission of 2 hours): the Correct Detection ratio equals 100%, the Missed Detection ratio and the False Alarm ratio equal 0.

Beyond that, the cost of time of training is favorable, because the weak classifiers don't need a deep machine learning and the training of weights of strong classifier is simple, as presented in previous section. It costs 16.4 minuets to complete all the machine learning with a computer - Intel Pentium Dual-Core T4400 and 2G memory.

4.3 Results of Test on the monitoring terminal off-line

The results of test on the monitoring terminal off-line are shown in Table 2. According to the requirements of engineersthe faults are divided into three sets. Thus, including the normal condition, there are four situations to distinguish by system. Clearly, the strong point of each subsystem is different. The ratio recognition (score) of subsystem BPLM is better than PNN, it can diagnose totally the fault 1 and 4, but it has 0.85% possibility to mistake fault 2 for Normal

Table 1. Testing results on the monitoring terminal on-line (CD:Correct Detection of fault,MD:Missed Detection,FA:False Alarm)

Classifiers	**CD**	**MD**	**FA**
Strong Classifier	99.00%	1.00%	4.29%
Weak Classifier 1	91.05%	8.95%	10.71%
Weak Classifier 2	90.49%	9.51%	13.57%
Weak Classifier 3	92.01%	7.99%	10.00%
Weak Classifier 4	91.45%	8.55%	12.86%
Weak Classifier 5	94.06%	5.94%	9.29%
Weak Classifier 6	95.12%	4.88%	7.86%
Weak Classifier 7	92.76%	7.24%	10.71%
Weak Classifier 8	92.06%	7.94%	11.43%
Weak Classifier 9	93.36%	6.64%	8.57%
Weak Classifier 10	93.16%	6.84%	8.96%

situation (Ref). Although subsystem PNN obtains less score than BPLM, it is a hundred per cent sure of the faults 1, 2 and 3, which can cover the shortage of BPML. But it also has a disadvantage that mistakes ref for faults (false alarm of faults 1 and 2).

The decision system, which comprehensive analyze the results of five subsystems and concludes a final diagnosis result. It is not a simple vote system. On basis of the results of test, rules of decision (thresholds of each output of each subsystem) are built to treat the results. For example, BPLM and GMM can well distinguish the normal situation, so their outputs of REF are more important than others. Again for instance that, BPLM and PNN are good at distinguishing the fault 3, so their outputs of fault 3 are treasured, which will multiply by a greater parameter. Finally, on the basis of complementary advantagesthe final decision is made as shown in the last part of Table 2. It is nearly a completely correct fault diagnosis.

5. Conclusions

In this paper, a fault diagnosis system of aircraft is built by Adaboost and five subsystems with different recognition methods. The whole system consists of two monitoring terminals on-line and off-line. At the monitoring terminal, Adaboost with its 10 weak classifiers based on BPLM supervises the situation of aircraft, send an alarm to pilot if there is a symptom of danger. At the monitoring terminal off-line, five subsystems cooperate with each other to deeply diagnose the faults and reach a conclusion of diagnosis for engineers and maintenance office. In addition to this, Since neural network, Decision Tree have the ability to do the machine learning of regression, possible future research includes predicting the aircraft faults based on the same structure of system.

Acknowledgements

All the data are supplied by Project MODIPRO "MOdelisation, DIagnosis and PROnostic".
Financiers : DGCIS, Conseil régional d'Ile de France, Conseils généraux 77 et 78.
Partners: DASSAULT AVIATION, ISIR/UPMC, IT4 CONTROL, INRIA, SupElec, SNECMA, BAYESIA, KBS.

Table 2. CONFUSION MATRIX OF TESTING RESULTS OF FIVE DIFFERENT SUBSYSTEMS (O : Outputs of systems, T : Targets of test)

T/O	REF	Fault 1	Fault 2	Fault 3
Results of Subsystem BPLM; Score = 96.77%				
REF	92.06%	0.79%	6.35%	0.79%
Fault 1	0.00%	100.00%	0.00%	0.00%
Fault 2	0.85%	0.00%	99.15%	0.00%
Fault 3	0.00%	0.00%	0.00%	100.00%
Results of Subsystem PNN; Score = 94.43%				
REF	84.92%	3.97%	0.11%	0.00%
Fault 1	0.00%	100.00%	0.00%	0.00%
Fault 2	0.00%	0.00%	100.00%	0.00%
Fault 3	0.00%	0.00%	0.00%	100.00%
Results of Subsystem LVQ; Score = 77.71%				
REF	71.43%	8.73%	19.84%	0.00%
Fault 1	0.00%	100.00%	0.00%	0.00%
Fault 2	0.85%	0.00%	99.15%	0.00%
Fault 3	20.51%	38.46%	41.03%	0.00%
Results of Subsystem GMM; Score = 97.65%				
REF	96.03%	0.00%	3.97%	0.00%
Fault 1	0.00%	100.00%	0.00%	0.00%
Fault 2	1.69%	0.00%	98.31%	0.00%
Fault 3	2.56%	0.00%	0.00%	97.44%
Results of Subsystem DT; Score = 93.84%				
REF	88.89%	2.39%	7.14%	1.58%
Fault 1	0.00%	100.00%	0.00%	0.00%
Fault 2	0.00%	3.39%	96.61%	0.00%
Fault 3	7.69%	0.00%	0.00%	92.31%
Final Decision System; Score = 99.70%				
REF	99.10%	0.65%	0.25%	0.00%
Fault 1	0.00%	100.00%	0.00%	0.00%
Fault 2	0.85%	0.00%	100.00%	0.00%
Fault 3	0.00%	0.00%	0.00%	100.00%

References

[1] A. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical systems and signal processing*, vol. 20, no. 7, pp. 1483–1510, 2006.

[2] R. Gao and A. Suryavanshi, "Bit for intelligent system design and condition monitoring," *Instrumentation and Measurement, IEEE Transactions on*, vol. 51, no. 5, pp. 1061–1067, 2002.

[3] J. Zhu, S. Rosset, H. Zou, and T. Hastie, "Multi-class adaboost," *Ann Arbor*, vol. 1001, no. 48109, p. 1612, 2006.

[4] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," in *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, 1996, pp. 148–156.

[5] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, p. 1612, 1999.

[6] W. Faller and S. Schreck, "Neural networks: applications and opportunities in aeronautics," *Progress in Aerospace Sciences*, vol. 32, no. 5, pp. 433–456, 1996.

[7] D. Specht, "Probabilistic neural networks," *Neural networks*, vol. 3, no. 1, pp. 109–118, 1990.

[8] R. Low and R. Togneri, "Speech recognition using the probabilistic neural network," in *Fifth International Conference on Spoken Language Processing*, 1998.

[9] L. Silva, A. Santos, A. Montes, and J. Simoes, "Hamming net and lvq neural networks for classification of computer network attacks: a comparative analysis," in *Neural Networks, 2006. SBRN'06. Ninth Brazilian Symposium on*, 2006, pp. 72–77.

[10] X. Gong, H. Meng, and X. Wang, "A gmm-based algorithm for classification of radar emitters," in *Signal Processing, 2008. ICSP 2008. 9th International Conference on*, 2008, pp. 2434–2437.

[11] G. Tauchen, "Diagnostic testing and evaluation of maximum likelihood models," *Journal of Econometrics*, vol. 30, no. 1-2, pp. 415–443, 1985.