Proceedings of the 7th WSEAS International Conference on Neural Networks, Cavtat, Croatia, June 12-14, 2006 (pp139-143)

Intelligent Digital Modulation Type Identifier

[1]ATAOLLH EBRAHIMZADEH, [2]SEYED ALIREZA SEYEDIN
Department of Electrical and Computer Engineering
[1]Noshirvani Institute of Technology, [1,2]Ferdowsi University of Mashad
IRAN

Abstract-Automatic modulation type identification is needed in many applications. Most of modulation type identification methods can only recognize a few kinds of signals. They usually require high levels of signal to noise ratio (SNR) to achieve an acceptable performance. This paper proposes a new intelligent digital modulation type identifier. This identifier uses a multilayer perceptron neural network with resilient back propagation learning algorithm as the classifier and higher order moments and cumulants (up to eighth) as the features. A validation method is used during training cycle to improve the generalization of the classifier. Genetic algorithm is utilized to finding the numbers of hidden layer nodes and selection of input features. The experiment results show that IDMTI is able to discriminate the different kinds of digital modulations with high accuracy even at very low SNR values.

Keywords: Neural network, statistical pattern recognition, modulation, optimization, higher order statistics.

1 Introduction

Automatic modulation type identification is an intermediate process between signal detection and demodulation. It has many communication intelligence applications such as: spectrum surveillance, interference identification, intelligent modems, software radios, etc. Whilst most methods proposed initially were designed for analogue modulations, the recent contributions in the subject focus more on digital communication due to increasing usage of digital modulations.

Generally, automatic modulation type identification methods fall into two main categories, decision theoretic (DT) and pattern recognition (PR). DT approaches use probabilistic and hypothesis testing arguments to formulate the recognition problem and to obtain the classification rule [1-3]. The major drawbacks of these approaches are their very high computational complexity and difficulty within the implementation of decision rule. PR approaches, however, are simple to implement. They can be further divided into two subsystems: the feature extraction subsystem and the classifier subsystem [4-11]. The selection of both subsystems is most serious problem. Multilayer perceptron (MLP) neural network is one of the classifier that is used in modulation identification systems. It is showed that this type of classifier outperforms other classifiers such as K-nearest neighborhood [7-11].

Most of modulation identification systems can only recognize only a few kinds of signals. The work presented here, proposes a new intelligent digital modulation type identifier (IDMTI) which is able to recognize different types of digital modulations from lower order to higher order even in low SNR values. Figure 1, shows the general scheme of IDMTI. Preprocessing module performs: rejection of noise outside of signal bandwidth, carrier frequency estimation (or to be known), recovery of complex envelope, etc. This stage is similar in most of methods and we don't explain more, here. The considered digital modulations set is presented in section 2. Feature extraction module is described in section 3. The classifier that is used in this paper is introduced in section 4. Section 5, presents an optimization problem that will be done by GA. Section 6, show some simulation results. Finally, section 7 concludes the paper.

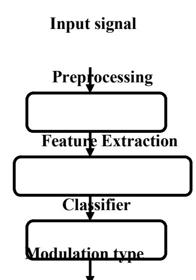

Figure1: General scheme of IDMTI

2 Considered digital modulations set

In digital communications, according to the changes in the message frequency, message amplitude, message phase, or changes in amplitude and phase, we have four main digital modulation techniques, frequency shift keying (FSK), amplitude shift keying (ASK), phase shift keying (PSK) and quadrature amplitude modulation (QAM), respectively. Most of them are applied in M-ary form because of their bandwidth efficiency [12]. We have:

$$S_{MFSK}(t) = \frac{\sqrt{2E_S}}{T_s} \mathrm{Re}\{\sum_k e^{j2\pi(f_c+\Delta f_k)t} g(t-kT_s)\}$$
$$\Delta f_k = \left[i - \left(\frac{M-1}{2} \right) \right] \Delta f, \quad i = 0,1,...,M-1 \tag{1}$$

$$S_{MASK}(t) = \frac{\sqrt{2E_S}}{T_s} \mathrm{Re}\{\sum_k A_k e^{j2\pi f_c t} g(t-kT_s)\}$$
$$A_k = 2i - M - 1, \quad i = 0,1,...,M-1 \tag{2}$$

$$S_{MPSK}(t) = \frac{\sqrt{2E_S}}{T_s} \mathrm{Re}\{\sum_k C_k e^{j2\pi f_c t} g(t-kT_s)\}$$
$$C_k = e^{j\frac{2\pi i}{M}}, \quad i = 0,1,...,M-1 \tag{3}$$

$$S_{MQAM}(t) = \frac{\sqrt{2E_S}}{T_s} \mathrm{Re}\{\sum_k C_k e^{j2\pi f_c t} g(t-kT_s)\}$$
$$C_k = a_k + jb_k ; a_k, b_k = 2i - M - 1, i = 0,1,...,M-1 \tag{4}$$

In the above equations $g(t)$ is pulse shaping function, f_c is frequency carrier, M is number of states, T_s is symbol period and E_s is energy per symbol. A_k, C_k and Δf_k states the symbols. Δf is frequency deviation. Among them, QAM types are a new type and as have a slightly different form. It is designed to transmit two separate signals independently with the same carrier frequency. In this paper the considered digital modulations set is {PSK2, PSK4, PSK8, ASK4, ASK8, QAM16, QAM32, QAM128, Star-QAM8, V29 (9600)}.

3 Features

In modulation identification problem, finding the proper features is very important. For example QAM modulation schemes contain information in both amplitude and phase (that are regarded as complex signals), thus finding the proper feature that could be able to identify them (especially in case of higher order and/or non-square) is difficult. Based on our researches, the higher order moments and higher order cumulants up to eighth achieve the most highly performances to discriminating of digital modulations such as considered set in this paper. These features have many advantages e.g. they provide a good way to describe the shape of the probability density function. Following briefly describe these features.

Probability distribution moments are a generalization of concept of the expected value [13]. The auto-moment of the random variable may be defined as follows:

$$M_{pq} = E[s^{p-q}(s^*)^q] \tag{5}$$

where p called moment order. Assume a zero-mean discrete based-band signal sequence of the form $s_k = a_k + jb_k$. Using the (5), the expressions for different order may be easily derived.

The symbolism for p^{th} order of cumulant is defined as:

$$C_{pq} = Cum[\underbrace{s,...,s}_{(p-q)terms}, \underbrace{s^*,...,s^*}_{(q)terms}] \tag{6}$$

There are relations between moments and cumulants. The n^{th} order cumulant is a function of the moments of orders up to including n. Moments is expressed in terms of cumulants as:

$$M[s_1,..,s_n] = \sum_{\forall v} Cum\left[\{s_j\}_{j\in v_1}\right]...um\left[\{s_j\}_{j\in v_1}\right] \tag{7}$$

where the summation index is over all partitions $v = (v_1,...,v_q)$ for the set of indexes $(1,2,...,n)$, and q is the number of elements in a given partition. Cumulants may be also be derived in terms of moments. The n^{th} order cumulant of a discrete signal *s(n)* is given by:

$$Cum[s_1,..,s_n] = \sum_{\forall v} (-1)^{q-1}(q-1)! E[\prod_{j\in v_1} s_j]..E[\prod_{j\in v_q} s_j] \tag{8}$$

We have computed the features of considered digital modulation set. For simplifying the indication, we substitute the modulations PSK2, PSK4, PSK8, ASK4, ASK8, QAM16, QAM32, QAM128, Star-QAM8, V29 (9600), with P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}, respectively. Table1 shows some of these features (values for signal constellations under the constraint of unit variance and noise free).

Table1: Some of features of considered modulations

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
M_{20}	1	0	0	1	1	0	0	0	0	0
M_{41}	1	0	0	1.64	1.77	0	0	0	0	0
M_{61}	1	-1	0	2.92	3.62	-1.32	-.38	.378	2.92	1.06
M_{82}	1	-1	0	5.25	7.92	-2.48	-.74	-.78	5.25	2.19
C_{40}	-2	-1	0	-1.36	-1.2	-0.68	-.19	-.18	1.64	.516
C_{63}	16	4	4	8.32	7.19	2.08	2.11	1.95	0.16	1.49
C_{80}	-244	-34	1	-30.1	9.27	-13.99	-1.9	-1.8	-88.9	-5.6

4 Classifier

MLP neural network is a popular family of feed forward ANNs that has simple and efficient applications [14]. In this paper, a MLP is used as the classifier. MLPs consist of an input layer of source nodes, one or more hidden layers of computation nodes (neurons) and an output layer. The number of nodes in the input and the output layers depend on the number of input and output variables, respectively. The number of hidden layers and the number of nodes in each hidden layer affect the generalization capability of the network. For smaller number of hidden layers and neurons, the performance may not be adequate, whereas with too many hidden nodes may have the risk of overfitting the training data and poor generalization on the new data. Figure 2 shows a typical MLP architecture consists of input, one hidden, output layers, respectively.

Inputs are propagated through the network layer by layer and MLP gives a non-linear mapping of the inputs at the output layers. The input vector $x=(x_1,x_2,...,x_N)^T$ is transformed to an intermediate vector of hidden variables **u,** using the activation function φ_1. The output u_j of the jth node in the hidden layer is obtained as follows:

$$u_j=\varphi_1\left(\sum_{l=1}^{N} w_{i,j}^{1} x_i + b_j^{1}\right) \tag{9}$$

where b_j^1 and $w_{i,j}^1$ represent the bias and the weight of the connection between the j^{th} node(in the hidden layer) and the i^{th} (input) node, respectively. The superscript 1 represents the connection (first) between the input and the hidden layers. The output vector $y=(y_1,y_2,...,y_Q)^T$ is obtained from the vector of intermediate variables **u** through a similar transformation using activation function φ_2 at the output layer. For example, the output of the neuron k can be expressed as follows:

$$u_k=\varphi_2\left(\sum_{l=1}^{M} w_{l,k}^{2} x_i + b_k^{2}\right) \tag{10}$$

where the superscript 2 represents the connection (second) between the neurons of the hidden and the output layers. There are several forms of activation functions φ_1 and φ_2, such as logistic function, hyperbolic tangent and piece-wise linear function.

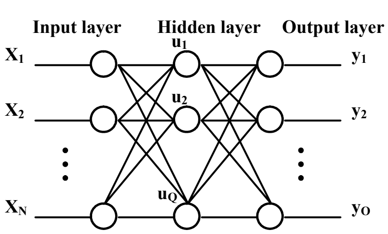

Figure3: Typical structure of MLP (one hidden layer)

The recognition basically consists of two phases training and testing. In training stage, weights are calculated according to the chosen learning algorithm. The training of MLP is very important. One of popular learning algorithm is standard back-propagation (BP) algorithm. However, in recent years, new learning algorithms have been proposed for network training. However, some algorithms require much computing power to achieve good training. In this paper, the resilient back-propagation algorithm (RPROP) is used [15].Unlike BPs, RPROP only considered the sign of derivatives as the indication for the direction of the weight update. In doing so, the size of the partial derivative does not influence the weight step. The following equation shows the adaptation of the update values of Δ_{ij} (weight changes) for the RPROP algorithm. For initialization, all Δ_{ij} are set to small positive values:

$$\Delta_{ij(t)}=\begin{cases}\eta^{+} * \Delta_{ij}(t-1); if & \dfrac{\delta E}{\delta w_{ij}}(t-1)\dfrac{\delta E}{\delta w_{ij}}(t) \succ 0\\ \eta^{-} * \Delta_{ij}(t-1); if & \dfrac{\delta E}{\delta w_{ij}}(t-1)\dfrac{\delta E}{\delta w_{ij}}(t) \prec 0\\ \eta^{0} * \Delta_{ij}(t-1); otherwise & \end{cases} \tag{11}$$

where $\eta^0=1$, $0 \prec \eta^- \prec 1 \prec \eta^+$ and $\eta^{-,0,+}$ are known as the update factors. Whenever the derivative of the corresponding weight changes its sign, it implies that the previous update value is too large and it has skipped a minimum. Therefore, the update value is then reduced (η^-) as shown above. However, if the derivative retains its sign, the update value is (η^+) increased. This will help to accelerate convergence in shallow areas. To avoid over-acceleration, in the epoch following the application of η^+, the new update value is neither increased nor decreased (η^0) from the previous one. Note that values of Δ_{ij} remain non-negative in every epoch. This update value adaptation process is then followed by the actual weight update process, which is governed by the following equations:

$$\Delta w_{ij}(t)=\begin{cases}-\Delta_{ij}(t) & if \quad \dfrac{\delta E}{\delta w_{ij}}(t) \succ 0\\ +\Delta_{ij}(t) & if \quad \dfrac{\delta E}{\delta w_{ij}}(t) \prec 0\\ 0 & otherwise\end{cases} \tag{12}$$

$$w_{ij}(t+1)=w_{ij}(t)+\Delta w_{ij}(t) \tag{13}$$

5 Genetic algorithm

The execution speed of a modulation type identifier is most important characteristic. One of the parameters that affects on the execution speed value is the number of features. As it is realized from the section

3, our modulation type identifier faces to a lot of features. Although whole of the features are useful but some of them are share the same information content. On the other hand, this number of features causes complexity computations. Thus it is needed to perform the feature selection. Another parameter that affects on the execution speed value of identifier is complexity of the classifier (MLP). The complexity of MLP is, usually, related to the number of hidden layer and nodes. The MLP that is considered in this paper has one hidden layer. Thus it should be only to reduce the number of hidden layer nodes. Both of the features selection and reduction nodes of hidden should be done in a manner without compromising the performance of the recognizer. This is an optimization problem. Among the methods that exist, it is considered the GAs method because of its efficiency and simplicity [16].

GA has been considered with increasing interest in a wide variety of applications. The use of genetic algorithm needs consideration of six basic issues: chromosome (genome) representation, selection function, genetic operators like mutation and crossover for reproduction function, creation of initial population, termination criteria, and the evaluation function. Though the traditional genome representation has been in binary form, the interest in real-coded or floating-point genomes for multidimensional parameter optimization problems is on the rise because of the closeness of the second type of representation to the problem space, better average performance and more efficient numerical implementation.

In this paper, real-coded genomes were used. The genome X contains $N+1$ real numbers $X=\{x_1 x_2 \ldots x_N x_{N+1}\}^T$ .The first N numbers in the genome are the selected features from the total features (R) and constrained to be in the range $1 \le x_i \le R$. The last number (x_{N+1}) shows the number of hidden neurons and has to be within the range $S_{\min} \le x_{N+1} \le S_{\max}$. The parameters $S_{\min}$ and $S_{\max}$ represent respectively the lower and the upper bounds on the number of nods in the hidden layer. A probabilistic selection, namely, normalized geometric ranking method [17] was used based on the individual's fitness such that the better individuals have higher chances of being selected. A non-uniform mutation function [18] using a random number for mutation based on current generation and the maximum generation number, among other parameters was adopted. Heuristic crossover [18] was used. This operator produces a linear extrapolation of two individuals using the fitness information.

To start the solution process, the random generation of initial solutions for the population is used. The maximum number of generations was adopted as the termination criterion for the solution process. The fitness function used here returns the number of correct classification of the test data.

6 Simulation study

All signal are digitally simulated according to (2), (3) and (4) in MATLAB simulation editor. Gaussian noise was added according to SNRs, –2, 0, 4, 8, 12, and 20 dB. Each modulation type has 3000 realizations of 4096 samples. These are then divided into data sets for training, validation and testing. The MLP classifier is allowed to run up to 4,000 training epochs. However, training is normally stopped by the validation process long before this maximum epoch is reached. The activation functions of tan-sigmoid (tanh), and logistic (log-sigmoid), were used in the hidden and the output layers, respectively. In this work, a MSE of 10E-6 is used.

Firstly we evaluate the performance of system without optimization (straight ANN), i.e., full features are used and the number of neurons in the hidden layer has been determined manually. Based on some simulations, the number of 17 neurons seems to be adequate for reasonable classification. Table 2 shows the performance of the identifier for various SNR values. Performance is generally good even with low SNRs.

Table 2: Overall performance (OP) of straight system

SNR (dB)	OP
-2	85.35
0	91.29
4	93.25
8	98.45
12	98.91
20	99.18

Now, we apply the GA. In the GA, a population size of ten individuals was used starting with randomly generated genomes. This size of population was chosen to ensure relatively high interchange among different genomes within the population and to reduce the likelihood of convergence within the population. The number of output layer node is set by the number of modulations. One has to specify the number of features (number of input nodes) that varies 3 to 28 (total number of features).Therefore, experiments were carried out to investigate the possibility of even smaller data set and the compromise in performance that one might observe. The number of hidden nodes varies between 4 and 24. The size of the hidden layer is determined by genetic algorithm itself during training. This allows training to proceed at a faster rate than an exhaustive training process that checks different sizes of first layer. The genetic algorithm is allowed to select subsets of various sizes to determine the optimum combination. Table 3 shows the performance of the identifier using only seven features selected (FS) by the GA. Also, at each SNR, the optimal number of neuron in hidden layer (ONHN) is identified. Results indicates that that for example at SNR= -2dB, the recognizer records a performance degradation of about 1.5% only. For other SNRs, the difference is negligible. Thus it can be said that the proposed method achieves high performance on most SNR values with only seven features and nearly 13 neurons that have been selected using GA.

Table3: Performance of identifier with applying of GA

SNR	No. FS	ONHN	OP with GA	OP without GA
-2	7	14	83.92	85.35
0	7	13	90.83	91.29
4	7	12	92.96	93.25
8	7	12	98.15	98.45
12	7	12	98.72	98.91
20	7	12	98.89	99.18

7 Conclusion

AMTI is an important issue in communication intelligence and electronic support measure systems. Here, we propose a new intelligent system (IDMTI) that uses high effective features and classier. It is able to discriminate the different kinds of digital modulations with high accuracy at very low SNR values. The proposed method is also fast in terms of training time.

Acknowledgements

The Authors would like to thank ITRC and TCKH.

References:

[1] C. Le Martret, and D. Boitea, A general maxi mum likelihood classifier for modulation classification, *Proc. ICASSP,* Vol. 4, 1998, pp. 2165-2168.

[2] W. Wei, and J. M. Mendel, Maximum-likelihood classification for digital amplitude-phase modulations, *IEEE Trans. Commun.*, 2000,Vol. 48, pp. 189-193.

[3] P. Panagotiou, and A. Polydoros, Likelihood ratio tests for modulation classifications, *Proc. MILCOM*, 2000, pp. 670-674.

[4] B. G. Mobasseri, Digital modulation classification using constellation shape, *Signal Processing,* Vol. 80, 2000, pp. 251–277.

[5] A. Swami, and B. M. Sadler, Hierarchical digital modulation classification using cumulants," *IEEE Trans. Comm.,* Vol. 48, No. 3, 2000, pp. 416–429.

[6] J. Lopatka, and P. Macrej, Automatic modulation classification using statistical moments and a fuzzy classifier, *Proc. ICSP, 2000*, pp.121-127.

[7] A. Ebrahimzadeh, S. A. Seyedin, A new method for identifying of signal type, Proc. CIS, 2005, 156-161.

[8] A. Ebrahimzadeh, and S. A. Seyedin, Automatic psk modulation identification using wpa and a modified mlp, Proc. CIS, 2005, pp. 250-255.

[9] C. L. P. Sehier, Automatic modulation recognition with a hierarchical neural network, Proc. MILCOM, 1993, pp. 111–115.

[10] A.K. Nandi, E.E. Azzouz, Algorithms for automatic modulation recognition of communication signals, IEEE Trans. Commun., Vol. 46, No. 4, 1998, pp. 431–436.

[11] L. Mingquan, X. Xianci, L. Leming, Cyclic spectral features based modulation recognition, Proc. Comm. Tech., Vol.2, 1998, pp. 792–795.

[12] J. G. Proakis, Digital Communications, New York: McGraw-Hill, 2001.

[13] P. McCullagh, Tensor Methods in statistics. Chapman & Hall, 1987.

[14] S. Haykin, Neural Networks: A Comprehensive Foundation. 2nd Edition, Prentice-Hall, NJ, USA, 1999.

[15] M. Riedmiller, H. Braun, A direct adaptive method for faster back-propagation learning: the RPROP algorithm, Proc. I CN N, 1993, pp. 586–591.

[16] G. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, NY, USA, 1989.

[17] Z. Michalewicz, Genetic algorithms+Data Structures=Evolution Programs. 3rd Edition, Springer, NY, USA,1999.

[18] C.R. Houk, J. Joines, M. Kay, A genetic algorithm for function optimization: a matlab implementation. North Carolina State University, Report no.: NCSU IE TR 95 09, 1995.