Working in a Robust Non-Linear Infant Cry Classifier

Sergio D.CanoOrtiz†, Daniel I.Escobedo Beceiro‡ and Taco Ekkel†

† University of Oriente, Cuba

‡ University of Twente, The Netherlands

E-mail: scano@fie.uo.edu.cu, daniele@fie.uo.edu.cu, taco@utwente.nl

ABSTRACT

The paper pretends to implement a parametric classification to lead to a robust parameter set which permit to find out the most efficient cry parameters for a robust classification of the cry units. It's achieved through an intelligent searching algorithm combined with a fast non-linear classification procedure, establishing the cry parameters which better matches the physiological status previously defined for the six control groups used as input data. Finally the optimal acoustic parameter set is chosen in order to implement a new non-linear classifier based on a radial basis function network, a neural NN-based procedure which classifies the cry units into a 2 categories, normal-or abnormal case.This study also represent one of the few attempts at the application of Artificial Neural Networks (ANN's) in the domain of infant cry as well as the first one oriented for diagnoses purpose on hypoxaemia-based diseases, at least on the documented results in the literature.

.Keywords: cry analysis, acoustics, neural network

1. INTRODUCTION

Last research has reported the effectiveness of cry analysis for the study of infant's neurological status. Since this early work, there have been a number of advances that have broadened the potential application of cry analysis not only for cry generation area but also for cry perception, in which the use of ANN's is one of the clue in the road [3,4,15,16]. However we know much about the cries of both healthy and sick infants, but a reliable investigation regime which can be used for clinical routine purposes, has yet not been developed

In this paper we address some general aspects concerned with the Golub's theory [11,17] and a computer-aided cry analysis is developed, in order to find out relevant aspects in the behaviour of cry parameters that correlate with medical abnormalities.

2. MATERIALS AND METHODS

The cry recordings were made at the Eastern Maternity Hospital of Santiago de Cuba. A pain cry was induced by a standardised stimulus: a ''heelstick'', the infants were positional decubito supine and flat in open cribs and were not crying at the time of the cry stimulus. The recordings were developed using a SONY CFS-210 tape recorder with a hand held PHILIPS SBC-3040 microphone, at a distance of approximately 10 cm from the baby's mouth. The average-time of the cries was 30 sec.The digitisation of the cry signals was made by an A/D acquisition system connected to a IBM-PC.

The data set consists of a successive cry episodes in which the average values for one cry unit is considered as a single pattern, resulting in 187 patterns to be evaluated [14]. The cry episodes were obtained from 35 neonates, nineteen healthy cases and sixteen sick neonates with hypoxia-based CNS diseases, considering six control groups of data (under the specialised criterion of physicians) as follow in Table 1:

Control Groups
Normal Group 1: normal transpelvian delivery
Normal Group 2:cesarean dysthocic
Pathologic Group 3: hypoxaemia
Pathologic Group 4: retarded intra-uterine growth
Pathologic Group 5: hypoxaemia with another risk
Pathologic Group 6: hyperbilirrubin

Table 1. The control groups.

Every pattern consists of 25 values as shown in Table 2,

Feature	Parameters
F_0 (mean, min., max., std dev , variability)	5
F_1 (mean, min., max., standard deviation , variability)	5
Voicedness (mean, min., max., std dev , variability)	5
Energy (mean, min., max., std dev, variability)	5
First Latency	1
Rising Melody tyoe	1
Stridor occurrences	1
Shift occurrences	1
Class	1

Table 2. Example results from the feature extraction process.

The goal of this research is to use crying features to automatically classify infant cries into those produced by children suffering from hypoxia-induced damage to the central nervous system, and those produced by children not suffering from this damage.

Since we assume there is a relationship between data and class membership we will assume that there are non-linear relationships. To find these, linear data analysis is insufficient and we must turn to non-linear methods such as neural networks [2,7,12].In practice, a huge amount of different neural network models exists, based on a wide variety of concepts and learning algorithms. Instead of describing every neural network model before making a

ISBN 1-876346-48-5 © 2003 UAB

choice, some properties typical to our application are considered in order to narrow down the number of available options.

- A set of features of 25 different variables that could serve as input are quite available. Intuitively, one might feel that the more features are used, the more information is available, and therefore, the better the network will function. In practice however, using too many features with a limited amount of training data inevitably leads to a phenomenon known as the *curse of dimensionality*.

The effect called *'curse of dimensionality'* causes the network to perform increasingly unsatisfactory as the ratio of the number of inputs versus the number of examples grows. A way of dealing with this problem is to *pre-process* the data in order to reduce the dimensionality.

Pre-processing

The goal of pre-processing in general is to present and /or transform the input data in such a way that it can be optimally processed by a neural network. A process called *feature selection* was developed: to try every possible feature that can be constructed from the whole feature set and then selecting the subset yielding the best results. The feature selection process was then optimized in order to make it manageable in terms of available resources.

Feature subset testing algorithm

Given the assumed nonlinearity in the data set, we are forced to discard linear discriminant analysis as a viable option for producing a performance criterion for a data set, and must turn to non-linear methods. When choosing a neural network model for testing subsets, it is important to choose one that has little manual parameters to be tuned. For example, some networks need a *learning rate* to be chosen when they are trained. This value can not or hardly be determined automatically, and therefore a number of different learning rates have to be tried to find the one yielding the best results. Obviously, this multiplies the total search time needed by the number of trials needed to find this optimal parameter. To counter this, we are looking for a fast, non-parametric, supervised network model with little or no parameters to tune.

This leads us to a class of neural networks known as *radial basis function networks.* These networks have their origin in a procedure known as the *exact interpolation procedure.*

Exact interpolation works as follows: consider a mapping from a d-dimensional input space x (consisting of N points) to a one-dimensional target space. An exact interpolation procedure introduces a set of *basis functions,* one for each data point x^n . The basis functions take the form of

$$\phi\left(\left\|x - x^n\right\|\right) \qquad (1)$$

where ϕ is some non-linear function depending on the distance between x and x^n. Usually, the Euclidean distance metric is used. Then, the functional mapping *h(x)* that is performed by the procedure is constructed by linearly combining the basis functions:

$$h(x) = \sum_n w_n \phi\left(\left\|x - x^n\right\|\right) \qquad (2)$$

Usually, the basis function ϕ is chosen to be the Gaussian

$$\phi(x) = \exp\left[-\frac{x^n}{2\sigma^2}\right] \qquad (3)$$

where σ is the 'width' of the basis function, a parameter controlling the smoothness of the total approximation *h(x).*

Many basis functions, similar to the Gaussian, will in fact work more or less satisfactory. The Gaussian is often chosen because of its analogy to the normal distribution, on which the probability density functions of many natural processes are based [7].

Exact interpolation creates a mapping that passes through all the data points in the data set, often resulting in severe overfitting. The goal of this exercise however is not to find the *best* approximator for the mapping at hand, but to find a *fast* approximator -sacrificing some accuracy if need be. So, as a part of our speed-optimization approach, we choose to ignore the possibly increased smoothness as a result of a non-linear optimization and select the described exact interpolation procedure for creating a mapping of a feature subset to the class output. The only parameter to be tuned in this algorithm is the width σ of the basis functions. This method is known as a Probabilistic Neural Network (PNN).

Taking advantage from the use of the matrix-oriented functionality of MATLAB, a fast algorithm written by Saku Kukkonen can do the construct-and-test-cycle in 0.04 s, about 25 times faster than the original PNN algorithm. Proper operation of the accelerated algorithm was tested on the well-known Iris data set that contains features of different kinds of flowers [9] and is often used as a benchmark data set for testing classifiers. The accelerated PNN algorithm reached a maximum correct classification of 99 % which is comparable to results found with other classifiers [10].

Search techniques

To account for the effects of interdependent variables , a better way is to start out with all features. Gradually single features are dropped from the set, each time that feature yielding the smallest decrease (or highest increase) in performance. Walking all feature subset lengths using this mechanism requires, for *n* features, only *n+(n-1)+...+1* subset evaluations.

Pre-processing results

In MATLAB , an algorithm was constructed for performing the search , using the classification rate of the accelerated PNN as criterion.

ISBN 1-876346-48-5 © 2003 UAB

Therefore, to select those features that are best *in general,* the whole feature search was repeated many times, each time with a different random partition of the pattern data into training- and test set. The results are then averaged, and those features that appear in optimal subsets most often can be considered the overall best features. The Table 3 shows the features that will be used according the feature search results:

Feature	Appears in best feature set
First latency	71%
Voicedness mean	60%
F_1 minimum	59%
F_1 mean	52%
F_0 mean	51%
Voicedness variability	41%
Energy maximum	30%
Energy maximum	30%
F_0 maximum	30%
Energy mean	26%

Table 3 The Best features

3. RESULTS AND DISCUSSION

Building the classificator

Normally, after the pre-processing stage, one would make the step to actual classification. In this case, classification has already been performed in the selection of the best feature set. However, as we optimized the classification algorithm for speed, there are probably some improvements that can be made in the classification stage. Apart from that, we should end up with an 'out-of-the-box'-classificator that can be used on new data, so this is the place to construct it.

We already discussed exact interpolation as a powerful classification algorithm which has a flaw, namely that it tends to over-fit the data set. The interpolation function yielding the best generalization is the one which is typically much smoother than the one that is the result of exact interpolation. To achieve this smoothness, the algorithm is adjusted so that the complexity of the underlying function, rather than the size of the data set, determines the number of basis functions. This is the notion on which the Radial Basis Function network is grounded:

(1) *The number M of basis functions is typically much less than the number of data points N. A training algorithm, rather than the data set itself, determines the basis function centers.*

(2*) Each basis function is given its own individual width rather than using a common width for all basis functions. These individual widths are also determined during training.The Radial Basis Function neural network as described above involves training, but as said before, there is room for increased network construction time since the feature set to be used is already determined.*

Using MATLAB's neural network, an RBF network was constructed to classify the data set resulting from the feature search. The data sets maintain their respective sizes in terms of number of patterns, however, the dimensionality of the sets has been reduced to 10.

First a simple classification algorithm based on a single RBF network is constructed. As with the feature selector however, the resulting network is quite unstable , so, just as in the feature selection process, a kind of averaging mechanism is devised. This time however, it is not enough to simply average the classification rate since the output itself is of critical importance in this version of the classifier (as opposed to the feature search algorithm where it is the classification rate that is the only important parameter- the actual results did not yet). Instead , a mechanism called bagging- shorthand for bootstrap aggregation- is introduced. Bagging is especially suitable for using in unstable classification where small perturbances in the data set can greatly influence results [1].

Bootstrap aggregation

Bootstrap aggregation is a form of *arcing:* adaptive re-weighting and combining. This general term refers to reusing or selecting data in order to improve classification [7]. In bagging, several versions of a training set, each creating by drawing samples with replacement from the data set, are created. Each training set is then used to train a classifier (RBF network). Then, in the classification stage, the outputs of all those networks are combined to form the final classification based on the vote of each individual network (also called component classifier). Since the classifier is now essentially composed of many neural networks, it is referred to as a *multiclassifier system*, and the individual networks are called *component classifiers.*

Sampling with replacement means picking one data point at random without removing it from the data set it came from, so that it may be selected again. A bootstrap sample from a data set may therefore contain the same data point more than once. By constructing many versions of the same data set, each component classifier will create a different 'view' on the mapping that's at the base of the classifier's data set.

In this multiclassifier system, all component classifiers are of the same type and share the same output format. The output of one component classifier may simply be seen as the probability that some case belongs in a certain class (or the probability that the case does not belong in the other class). We may therefore simply linearly add the probabilities provided by the component classifiers and average them. After this , rounding the average value directly provides us with the classification result.

A multiclassifier system of 10 networks was constructed. The performance on the data set was 85%. Confusion matrices are shown in Table 4, they show a slight inclination towards type 1 errors ('false alarms') which is a 'good' thing (in medical terms, a false alarm could be

ISBN 1-876346-48-5 © 2003 UAB

considered less damaging than unjustly *not* reporting an abnormality). The resulting networks are 'fixed' by writing them to a MATLAB variable store so they can be used in real classification tasks.

Actual class	Group size	Predicted class	
		0	1
0	15	79%	21%
1	21	12%	88%

Table 4 Confusion matrix for multiclassifier system .

(0: normal, 1: abnormal)

The classifier's performance (correct classification rate) was 85% (with +/-5% on the confidence interval).

4. CONCLUSIONS

The extraction and classification system that was devised was capable of distinguishing between normal and abnormal cases with up to 85% (with +/-5% on the confidence interval). This result can be regarded as statistically significant when you compare with recent works [4,16].

It is interesting to see that, as compared to previous investigations in classification of hypoxia-related infant cries , especially the newly deployed features such as voicedness, energy and latency are most critical for reaching this classification rate. It is assumed that these features, coupled with the deployment of kernel-based neural networks for both feature selection and classification, made it possible to get to the aforementioned classification rate. The classification performances that were found quantify the network´s ability to relate some examples of measured values to some examples of diagnosed classes. Because of the limited set of samples from a real life process, they do not necessarily describe the relation as it is in real life.

Although the results presented here are preliminary in nature, more tests are currently ongoing. Different features for input into the neural network and other network architectures are currently being investigated so that it can be determined which features are the most suitable for correct classification of cry units

REFERENCES

[1]. Breiman L., *Bagging Predictors,* Journal of Machine Learning, vol. 24 no. 2 (1996), pp123-140

[2]. Bishop C., *Neural Networks for Pattern Recognition,* Oxford University Press, 1995, ISBN 019853864 2

[3]. Cano S. D., Escobedo D. l., *Spectral Analysis of the Infant* Cry: An *initial approximation, EUROSPEECH* 95 proceedings, Madrid, 1995

[4]. Cano S. D., Escobedo D. l., *Clasificación* de *unidades* de *llanto infantil mediante el mapa auto-organizado* de *kohonen,* AIRENE, Grupo de Procesamiento de Voz, Universidad de Oriente, Santiago de Cuba, Cuba, 1998

[5]. Cano S. D., Socarras M., Escobedo D. l., *Las técnicas* de *corte duración* en *el procesamiento digital* de *la* voz, Memorias V Congreso Latinoamericano de control automatico, Ciudad de La Habana, Cuba (1992)

[6]. Diez H., Torres M., Escobedo D. l., Regueiferos L., Capdevila L., *Diagnóstico del llanto infantil: una evaluación* de *clasificación usando técnicas* de *Reconocimiento* de *Patrones,* Universidad de Oriente, Santiago de Cuba, Cuba (1996)

[7]. Duda R., Hart Po, Stork D., *Pattern Classification,* 2 *nd edition,* John Wiley & Sons, Inc., ISBN 0-471-05669-3, 2001.

[8]. Escobedo D., Cano S., Collo E., Regueiferos L., Capdevila L., *Rising shift of pitch frequency in the infant* cry *of some pathologic cases,* 2nd International workshop on Models and Analysis of Vocal Emissions for Biomedical Applications, MA VEBA 200l, Firenze, Italy 2001.

[9]. Fischer R., *The* use *of multiple measurements in taxonomic problems,* Annals of Eugenics 7 (1936), pp. 179-188.

[10]. Ghahramani Z., Jordan M., *Learning from incomplete data,* A.I. Memo No. 1509, MIT, 1994.

[11]. Golub H., Corwin M., *Infant cry: a clue to diagnosis,* Pediatrics, vol. 69 (1982), pp. 197-201.

[12]. Haykin S., *Neural Networks: a comprehensive foundation, ed.2,* Prentice-Hall Inc., 1999.

[13]. Hirschberg J., *Dysphonia in infants,* International Journal of Pediatric Otorhinolaryngology 49 suppl.1 (1999), pp. S293-S296

[14]. Mulder M., *Infant* Cry *Analysis: a clue using self-organizing maps,* University of Twente, Faculty of Computer Science, Department of Language, Knowledge and Interaction, Enschede, Netherlands, 2001

[15]. Robb M., Cacace A., *Estimation of formant frequencies in infant cry,* International Journal of Pediatric Otorhinolaryngology 32 (1995), pp. 57-67

[16]. Schonweiler R., Kaese S.,Mo11er S., Rinscheid A., Ptok M., *Neuronal networks and self- organizing maps: new computer techniques in the acoustic evaluation of the infant* cry, Intemational Journal of Pediatric Otorhinolaryngology 38 (1996), pp. 1-11

[17]. Wasz-Hockert O., Lind J., Vuorenkoski V., Partanen T., Valanne E., *The infant* cry: *a spectrographic and auditory analysis,* Clin. Devo Med. 29 (1968) pp. 1-42.