MEDINFO 2007
HISA Ltd Australia
© 2007 The authors. All rights reserved.

On Classification of Otoneurological Cases Based on Vestibulo-ocular Reflex Signals

Martti Juhola[a], Heikki Aalto[b], Timo Hirvonen[b]

[a]*Department of Computer Sciences, University of Tampere, Tampere, Finland*
[b]*Department of Otorhinolaryngology, Helsinki University Central Hospital, Helsinki, Finland*

Abstract

According to signal analysis and pattern recognition results earlier computed by us, we distinguished a set of patients from the healthy subjects. Classification was executed on the basis of machine learning methods to explore, which methods are the most efficient to identify the two classes from each other. We implemented programs for nearest neighbour searching, clustering, decision trees, Bayesian decision rule, multilayer perceptron networks and Kohonen networks. The last method was better than the others and obtained total accuracies as high as 94 %.

Keywords:

vestibulo-ocular reflex, eye movements, classification

Introduction

In otoneurology diseases entailing vertigo and other balance problems are investigated, among others, with signal analysis techniques applied to various measurements recorded from subjects. We began our research on this topic already in the 1980s by using syntactic pattern recognition methods for saccades, nystagmus, sinusoidal tracking eye movements and vestibulo-ocular reflex.

We constructed a computer program developed from our earlier research to recognize vestibulo-ocular reflex responses. In this research we continue the work by testing and surveying several classification methods to identify patients from healthy subjects along with the signal analysis results given by our syntactic pattern recognition technique.

Methods

In order to yield vestibulo-ocular reflex, head movements have to be made. We have developed a computer-controlled mechanical device to produce carefully guided horizontal head movements of a subject. For this purpose, a subject sat in a fixed chair and a boxer's helmet was tightly set on the subject's head. Two push rods controlled by an electronic DC motor and connected to the boxer's helmet moved the head from the centre to the left or right and back after an interval of 0.7-1.2 s.

During the measurement the subject was due to keep the gaze toward a fixed object. Each head movement stimulates a rather symmetric eye movement response in the opposite direction. Head movements were recorded with a potentiometer. Both signals were measured at the same time at 400 Hz and lowpass filtered to dampen noise above approximately 70 Hz. Signals of 22 healthy subjects and 22 patients were measured for 80 s.

We applied our syntactic method to recognize stimulation head movements and their eye movement response from each signal pair. Latency and gain values between stimulations and responses were computed. We computed three gain estimates. The first one, g_1, was computed according to linear regression. Second, we applied the ratio g_2 of the maximum velocity values of the response and stimulation and the ratio g_3 of their mean angular velocities. All three gains were computed on the basis of the 100 ms duration immediately after the beginning of a vestibulo-ocular reflex.

We computed average gain and latency values for the group of the 22 randomly selected healthy control subjects and similarly for the 22 patients, who suffered from unilateral otoneurological dysfunctions caused primarily by an operated acousticus neuroma. We separated their intact sides from those of the dysfunction since the intact side should be almost normal.

Results of machine learning methods

Our goal was to distinguish the healthy subjects from the patients by means of algorithms implemented in Matlab. They were run to compare their efficacy in the classification based on eye movements of vestibulo-reflex. We applied ten-fold crossvalidation by dividing the stimulation-response pairs of both subject groups (483 from the healthy and 371 from the patients) into ten subsets. If a method was such as neural networks that had some random initialization values, ten runs were still performed for each crossvalidation situation. Sensitivity, positive predictive value and total accuracy were computed for both classes.

Tests of *k*-nearest neighbour searching (Table 1) and *k*-means clustering in the Euclidean variable space (Table 2)

were run. We tested decision trees (Table 3), Bayesian decision rule (Table 4) and multilayer perceptron neural networks (Table 5).

Table 1 - Means and standard deviations [%] of classification results with k-nearest neighbour searching

k	**Sensitivity**		**Positive predictive value**		**Total accu-racy**
	healthy	**disor-dered**	**healthy**	**disor-dered**	
1	86.9±7.2	80.5±12.7	85.9±8.3	82.7±8.0	84.2±7.5
3	93.8±5.6	76.2±13.4	84.3±7.6	90.7±6.7	86.2±6.4

Table 2 - Means and standard deviations [%] of classification results with k-means clustering

k	**Sensitivity**		**Positive predictive value**		**Total accu-racy**
	healthy	**disor-dered**	**healthy**	**disor-dered**	
2	90.2±7.2	78.2±10.7	84.8±6.7	86.6±8.6	85.0±6.2
4	63.1±15.9	48.8±14.8	61.3±10.1	51.3±11.5	56.9±10.7

Table 3 - Means and standard deviations [%] of classification results with decision trees

Mean of leaves used	**Sensitivity**		**Positive predic-tive value**		**Total accuracy**
	healthy	**disor-dered**	**healthy**	**disor-dered**	
25.9	94.7±6.0	83.6±13.3	89.0±7.8	92.6±7.3	89.8±6.9

Table 4 - Means and standard deviations [%] of classification results with Bayesian decision rule

Sensitivity		**Positive predic-tive value**		**Total accuracy**
healthy	**disor-dered**	**healthy**	**disor-dered**	
92.5±7.0	82.7±11.3	87.9±7.3	89.8±8.8	88.3±7.1

Table 5 - Means and standard deviations [%] of classification results with multilayer perceptron neural networks

Hid-den nodes	**Sensitivity**		**Positive predic-tive value**		**Total accu-racy**
	healthy	**disor-dered**	**healthy**	**disor-dered**	
4	94.3±6.7	80.6±14.5	87.1±7.8	91.0±12.5	88.4±7.5
8	93.4±7.9	82.0±11.4	87.5±7.1	91.1±9.5	88.5±7.3
16	93.4±8.2	82.9±11.6	88.1±7.1	91.2±9.3	88.8±7.3

Kohonen networks were tested with ten runs with hexagonal neighbourhood pattern used with link distance (Table 6). Networks including 36 or more nodes reached 3-11 % higher outcomes than those in Tables 1-5.

Table 6 - Means and standard deviations [%] of classification results with Kohonen neural networks

Nodes	**Sensitivity**		**Positive predic-tive value**		**Total accu-racy**
	healthy	**disor-dered**	**healthy**	**disor-dered**	
3 × 3	94.4±4.8	83.5±9.6	88.6±6.3	92.3±6.4	89.7±4.9
5 × 5	94.8±2.2	89.5±9.6	92.5±6.4	93.0±2.9	92.5±4.4
7 × 7	98.1±2.3	90.0±7.2	93.0±4.8	97.5±3.1	94.6±3.4

Using Wilcoxon matched-pairs signed-ranks test, the efficiency of the methods are compared to each other. Because the *k*-nearest neighbour searching method, Bayesian decision rule and Kohonen networks were tested with ten and the others with 100 test settings, we compare inside these two method groups. The results of Kohonen networks of size 7 × 7 significantly differed from those of the *k* nearest neighbour searching of *k*=3 at the significant level of 0.001 and from those of the Bayesian decision rule at 0.01. The results of the perceptron networks of size 16 hidden nodes highly significantly differed from those of the *k*-means clustering with *k*=2. Further, the results of the decision trees highly significantly differed from clustering results and perceptron networks at 0.001.

Discussion and conclusion

The methods of *k*-nearest neighbour searching and *k*-means clustering produced average total accuracies of 84-86 %, the Bayesian decision rule evolved 88 % and the decision trees gave as high as 90 % total accuracies. Per-

ceptron neural networks obtained 88 % total accuracies. Thus, the nearest neighbour searching and clustering were the weakest, the multilayer perceptron networks followed with the Bayesian decision rule, the decision tree method was the next best, and the Kohonen neural networks were the best. The running times (with the 3 GHz processor) expectedly favoured the first three simple methods. The approximate durations were 7 s for the 10 nearest neighbour searching tests, 3 s for the 100 k means clustering tests, 1 min 13 s for the 100 decision tree tests. The Bayesian decision rule was the fastest, with clearly less than 1 s. For the 100 tests of the multilayer perceptron networks they were approximately 4 min 25 s for two hidden nodes and 6 min 23 s for 16 hidden nodes. For the 10 tests of the Kohonen neural networks they were 2 h 15 min for 2 × 2 nodes and 2 h 21 min for 7 × 7 nodes. These durations show how more effective but also complicated methods require considerably more time. On the other hand, the training process of the networks only once run engulfed most of the running time.