POLITECNICO DI TORINO

Master of science in ICT for smart societies

Report on ICT for Health Laboratory N°4

Hidden Markov Machines and Parkinson's disease

Author:
Bruno Valente

a.a. 2018-2019

1 Introduction

Parkinson's disease (PD) is a progressive degenerative disorder of the central nervous system caused by a loss of dopaminergic neurons in a part of the brain called substantia nigra. The symptoms of PD usually develop gradually and can affect both motor and non-motor functions. The main physical symptoms that characterise this condition are tremor, slowness of movements (bradykinesia) and muscle stiffness or rigidity, along with difficulty in starting an action (akinesia) and balance problems. In terms of cognitive and psychiatric impairments, patients with PD can experience anxiety, depression or apathy and cerebral deterioration. Patients' quality of life can also be affected by other features, such as mood swings, pain, sleep disorders, dementia and other mental health issues. Some people can show different kinds of **speech** and **communication impairments**, including slurring and mumbling, hoarseness, weak or strained voice, cluttering and problems related to the rhythm, tone and volume of speech.

The goal of this laboratory is to classify voice recordings as belonging to healthy people or people affected by PD using **Hidden Markov Machines (HMMs)**. The dataset consists of 20 audio files: 10 for healthy control (HC) subjects and 10 for PD patients. Each file stores the recording of the vowel "a" pronounced for some seconds by the 20 individuals above described.

2 Data pre-processing

The voiced speech of a person has fundamental frequencies belonging to different frequency ranges, depending on age, sex, etc. of the individual (e.g average male voices have frequencies in the range $70 - 200\ Hz$, while for women they can reach $400\ Hz$).

The idea is to find the **fundamental frequency** (f_0) for each voice signal; **resample** it with a given number of samples (N_s) in each period; and **quantize** it using N_q values. In order to achieve the task, we first select a 2 seconds interval in the middle of the recording and evaluate its Fourier transform. The peak at the lowest frequency gives f_0. Each "period" (values between two peaks at distance approximately equal to $1/f_0$) in time domain is interpolated using $N_s - 1$ equally spaced samples. Then, the K-means algorithm is run separately for HC and PD training signals and two sets of centroids (i.e. quantized levels) are obtained. At the end, all the voices are quantized.

3 Classification using HMMs

The main idea is to train **two different HMMs** with N_s states and N_q output values: one for healthy control people, one for Parkinson's disease patients. Then, giving a signal as input to both HMMs, it is possible to evaluate the probability that the signal belongs to the first or the second HMM. The largest probability will determine the class (HC or PD) the voice belongs to. The **training set** will consist of 7 healthy voices and 7 PD voices, the remaining signals will be used as **testing**

set. In addition, a sort of 3-fold cross validation will be performed, using 3 different subsets of signals for testing (and the remaining for training):

- *Subset 1*: signals 8, 9, 10 as test data;
- *Subset 2*: signals 1, 2, 3 as test data;
- *Subset 3*: signals 5, 6, 7 as test data;

The algorithm used in order to train the HMMs is the **Baum-Welch algorithm**. It requires as parameters: starting transition (A) and emission (B) matrices, the tolerance used for testing convergence of the iterative estimation process and the maximum number of iterations. It outputs the estimated transition and emission matrices for the Hidden Markov Model.
Finally, **specificity** and **sensitivity** will be measured for both training and testing signals. It will be interesting to play with the input parameters of the Baum-Welch algorithm and with the value of N_s and N_q, in order to see how specificity and sensibility vary.
Unless otherwise specified, the default values of the parameters involved in the analysis will be the following: $N_s = 8$, $N_q = 8$, tolerance equal to 10^{-3}, and the max number of iteration set to 200.

3.1 Different starting transition matrix

The elements of the initial emission matrix are picked at random. Each of them is divided by the sum of the elements of the row it belongs to, making sure that the matrix is stochastic.
Two options for the starting transition matrix has been analysed:

1. The initial transition matrix is a **circulant matrix**, defined as follows:

$$A = \begin{bmatrix} q & p & q & q & \cdots & q \\ q & q & p & q & \cdots & q \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p & q & q & q & \cdots & q \end{bmatrix}, \text{ where } q = \frac{1-p}{N_s - 1}$$

   The value of p is arbitrary and is set to 0.9 in our analysis. It represents the probability of moving from state n to state $n+1$ (modulo N_s) (e.g. from state 1 to state 2, from state 2 to state 3, ..., from state N_s to state 1).
   The value of q is defined such that the matrix is guaranteed to be stochastic (each row sums to 1). It represents the probability of moving from state n to any other state (including itself), except $n + 1$.

2. The elements of the initial transition matrix are picked at **random** and normalized in the same way it has been done for the emission matrix.

The significant difference between the two methods is that the first is a lot less time-consuming for the algorithm to converge because it requires fewer iterations. In fact, using the second initialization for the transition matrix, the elapsed time for the training part is 4-5 times higher than using the first method.

3.2 Different N_s

Using default parameters $(N_s = N_q = 8)$ we obtain the results in Table 3.1, where y_{test} contains the true classes, $\hat{y}_{test}$ the estimated ones. In red the wrong estimations.

y_{test}	HC	HC	HC	PD	PD	PD
$\hat{y}_{test}$	PD	PD	PD	PD	PD	PD

(a) *Subset 1*

y_{test}	HC	HC	HC	PD	PD	PD
$\hat{y}_{test}$	HC	HC	HC	PD	PD	PD

(b) *Subset 2*

y_{test}	HC	HC	HC	PD	PD	PD
$\hat{y}_{test}$	HC	HC	HC	HC	PD	PD

(c) *Subset 3*

Table 3.1: Comparisons between y_{test} and $\hat{y}_{test}$

The values of specificity are: 0 for *Subset 1*, 1 for *Subset 2* and *Subset 3*.
The values of sensitivity are: 1 for *Subset 1* and *Subset 2* and 0.667 for *Subset 3*. Hence, the overall **specificity** for the test set is 0.667, while the overall **sensibility** is 0.889.
Tables with the comparisons between true classes and estimated ones for the training signals (used as if they were testing data) will not be provided. However, the overall specificity and sensibility obtained for training signals are 1 and 0.952 respectively. They are, as expected, higher than the ones obtained with the test set.

For $N_s = 11$ the results in Table 3.2 are obtained.

y_{test}	HC	HC	HC	PD	PD	PD
$\hat{y}_{test}$	PD	HC	HC	PD	PD	PD

(a) *Subset 1*

y_{test}	HC	HC	HC	PD	PD	PD
$\hat{y}_{test}$	HC	HC	HC	PD	PD	PD

(b) *Subset 2*

y_{test}	HC	HC	HC	PD	PD	PD
$\hat{y}_{test}$	HC	HC	HC	HC	PD	PD

(c) *Subset 3*

Table 3.2: Comparisons between y_{test} and $\hat{y}_{test}$

Looking at Table 3.2, the overall **specificity** and **sensitivity** for testing signals are soon evaluated: 0.889 for both the statistical measures.
The values of specificity and sensitivity for training voices are 0.952 and 1, respectively.

3.3 Different N_q

Setting N_q equal to 9, classifier's performance improves, compared to the "default model". Results are reported in Table 3.3.

y_{test}	HC	HC	HC	PD	PD	PD
$\hat{y}_{test}$	HC	PD	HC	PD	PD	PD

(a) *Subset 1*

y_{test}	HC	HC	HC	PD	PD	PD
$\hat{y}_{test}$	HC	HC	HC	PD	PD	PD

(b) *Subset 2*

y_{test}	HC	HC	HC	PD	PD	PD
$\hat{y}_{test}$	HC	HC	HC	HC	PD	PD

(c) *Subset 3*

Table 3.3: Comparisons between y_{test} and $\hat{y}_{test}$

Overall **specificity** and **sensitivity** for testing signals are: 0.889 for both the statistical measures.
For training set, instead, the value of specificity and sensitivity is 1.

For some values of N_q the K-means algorithm fails to converge in the given max number of iteration. Therefore, it is important to find a compromise between the parameters involved in the model, in order to achieve reasonable results in a reasonable time.