ICT for Health
Laboratory # 4
HMM

Monica Visintin

Politecnico di Torino

2018/19

Table of Contents

Parkinson's Disease

People affected by Parkinson's Disease (PD) might have voice disorders, since they cannot properly control their muscles.
Goal of this laboratory is to train two Hidden Markov Machines (HMMs), one for healthy people and one for people affected by PD, so that it is possible to classify a voice as either normal or pathological.
Ten patients affected by PD were asked to pronounce "a" for a long time (according to their possibility) and the sound was recorded using a mobile phone with sampling frequency equal to 8 kHz. Healthy people with similar age were also asked to record their voices. The input dataset is therefore made of 20 wav files, 10 for healthy control (HC) and 10 for PD patients.
Matlab will be used for this lab. One function (`pre_process_data.m`) and the initial part of the main script (`main.m`) are provided, together with the wav files in the zipped file `HMM_for_students.zip` that can be downloaded from folder `materiale` of the class webpage.

Table of Contents

Description of function `pre_process_data` [1]

- Pronunciation of a vowel is obtained by pumping air out of the lungs, while the vocal cords periodically open and close the vocal tract.
- The frequency of vibration of the vocal cords is the **fundamental frequency** f_0 and actually varies from person to person (higher for females and children, lower for men). The idea is to find the fundamental frequency for each signal $x(t)$, and take exactly N_s samples in each period:
  - FFT is used to estimate the Fourier transform of the voice signal, characterized by several peaks at frequencies f_0 and its integer multiples; the peak at the lowest frequency gives the fundamental frequency f_0.
  - The signal in the time domain is then analyzed and its peaks are found at distance (time interval) approximately equal to $1/f_0$ (the signal is not exactly periodic).
  - Each interval between two peaks is interpolated taking $N_s - 1$ samples between the two peaks

Description of function `pre_process_data` [2]

The newly obtained signal $x[n]$ has period equal to N_s samples whatever was the value of f_0 (but it is not exactly periodic due to amplitude variations).

- Signal $x[n]$ is then quantized using N_q values, obtaining signal $x_q[n]$. Some of the 10 signals are used for training and the remaining are used for testing. The training signals are used to train the K-means algorithm (in this case it would be more appropriate to call this as Max-Lloyd algorithm) and find N_q centroids (i.e. representative levels). Then all the signals (both the training and the testing ones) are quantized (each sample is given to the cluster whose centroid is at minimum distance). The K-means algorithm is run separately for HC and PD voices, two sets of centroids are obtained.

Description of function `pre_process_data` [3]

- Function `pre_process_data` outputs two cell vectors (i.e. two lists of arrays), one (`hq`) with the 10 re-sampled and quantized voices of the healthy people and one (`pq`) with the 10 re-sampled and quantized voices of the PD patients. The cell vectors are organized so that they can be directly processed by the HMM functions in Matlab.

What is described above was already implemented in function `pre_process_data` and it is described to let you understand the entire system. Your tasks are described in the next slides. However you are invited to read file `pre_process_data.m` and understand it.

What you have to do [1]

- The idea is that if the signals were exactly periodic, then the signal could be generated by a finite state machine with N_s states: from state k the machine moves to state $k+1$ (modulo N_s) and when the machine is in state k it outputs the quantized level k.
- Since the signals are not exactly periodic, it is necessary to **train** the finite state machine, that has N_s states and N_q output values, in order to get the true transition and emission matrices. You need to train two HMMs: one for healthy voices, one for PD voices.

What you have to do [2]

- In the script `main.m` the parameters are set as $N_s = 8$ (`Nstates`), $N_q = 8$ (`Kquant`), but other values are possible.
- The first task is then to use Matlab function `hmmtrain` to get the transition and emission matrices. It is suggested to use the Baum-Welch algorithm (the Viterbi algorithm is not properly implemented and sometimes it crashes) with tolerance 10^{-3} and maximum number of iterations equal to 200 (other values are possible, but these are sufficient). The algorithm needs the **starting matrices** and **two options must be analyzed** for the transition matrix:
  1. The first row of the initial transition matrix is $[p, q, q, q, \ldots, q]$ with $q = (1-p)/(N_s - 1)$, and the matrix is circulant (the second row is the first one circularly shifted to the right by one position, etc)
  2. The initial transition matrix is generated using function `rand` (that generates random numbers between 0 and 1) and appropriately normalizing the result

  The emission matrix has to be set randomly (again use `rand` and normalize the obtained matrix so that it is stochastic).

What you have to do [3]

- Note again that **two HMMs have to be trained**, one for healthy people (HC-HMM), one for PD people (PD-HMM). The two HMMs have the same number of states and quantized output levels, but have different transition and emission matrices.
- Then use Matlab function `hmmdecode` to **check the results**. In particular, each of the 20 signals (including those used for training) must be given as input to both HMMs. Function `hmmdecode` outputs the probability that the input signal belongs to that machine (specified by the emission and tranision matrices). Then for each signal $x_q[n]$ you have two probabilities: if the largest probability is that related to the PD HMM, then the voice is classified as PD, otherwise it is classified as healthy. The estimated and true classes of the 20 signals can be compared to measure **specificity and sensitivity**. Don't mix signals used for training with signals used for testing (give sensitivity and specificity for the training signals, and sensitivity and specificity for testing signals).

What you have to do [4]

- In the script `main.m` signals from 1 to 7 are used for training (`ktrain`), signals from 8 to 10 for testing (`ktest`). Check your results also when the testing signals are
  - from 1 to 3
  - from 4 to 7

  (sort of 3-fold cross validation).

What you have to do [5]

In the overall:

1. Measure sensitivity and specificity of the suggested method to classify voices as PD or healthy
2. Use 3 different subsets of patients for training
3. Use two different initializations for the transition matrix
4. Play with parameters N_s and N_q and see if there are constraints on these values to get reasonable results.
5. Play with parameters tolerance and maximum number of iterations of `hmmtrain`

Comment your results in the report.

Report

- Report due by January 27th 2019 (Sunday) at midnight, upload it in folder "elaborati" of the class webpage
- Maximum number of pages for the report is 4 (concise and complete)
- Remember to:
  - use filename `yoursurname_sxxxxxx_report_4.pdf` for the report and filename `yoursurname_sxxxxxx_report_4.zip` for the zipped file that includes the pdf report and all the Matlab files; `xxxxxx` is you matricola number, `yoursurname` is your surname (otherwise the grade is 0 out of 5)
  - use the spellchecker
  - number the figures and write the caption (the figure title is NOT the caption)
  - refer to the figures in the text using the figure number
  - number the pages and justify your text
  - use sections, subsections, etc
  - read your report 3 times before you submit it
- Your report must be understood by a professor who teaches machine learning but does not know our notation, does not know about the dataset, does not know what the lab is about. However it is not necessary that you describe function `pre_process_data` in detail.