Machine Learning
Chapter 3: Hidden Markov models

1 Introduction

The purpose of this assignments is to understand and program the basics of the Baum-Welch algorithm to train a Hidden Markov model, which is based on the Expectation Maximization algorithm seen in the previous chapter. For Sections 3 and 4, assume that the transition probability matrix and the Gaussian parameters are known. In Section 5 you will be asked to produce estimates of these parameteres.

2 Data generation

The following code implements an HMM with a number K of hidden states z_n and an univariate random variable x whose mean and variance depend on the value of the hidden state. The inputs are the transition probability matrix P, vectors σ and μ containing the standard deviations and the means of the distributions, and the number of observations.

```
function [x,z]=markovprocess(P,sigma,mu,N)
p=cumsum(P,2);
zact=ceil(rand*length(mu));
z=[];
for i=1:N
    a=rand;
    zact=[min(find(p(zact,:)>a))];
    z=[z zact];
end
x=randn(size(z)).*sigma(z)+mu(z);
x=x'
```

Run the function and represent 100 hidden states and observations for a transition matrix

$$P = \begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0.3 & 0.1 & 0.6 \end{pmatrix}$$

and means and standard deviations $\mu = [1, 2, 3]$ and $\sigma = [0.3, 0.3, 0.3]$. Find an estimate of the transition probability matrix just counting the transitions.

3 Forwards step

Write and execute a code that computes the forwards algorithm in matrix form, this is, that iteratively computes the values of $\alpha_n(j)$ for all time instants n and all possible states j. Represent the result and compare it with the actual values of the transitions.

4 Backwards step

Write and execute a code that computes the backwards algorithm, this is, that computes the values of $\beta_n(j)$. In order to check it, compare it with the adequate variable of the script in section 2.

5 Forwards-backwards

Implement the fordwards-backwards algorithm and graphically test that the algorithm outputs the right result.

6 The Baum-Welch algorithm

Use the previously constructed scripts to implement the Baum-Welch algorithm. First, implement a function that computes matrices $\xi()$. Then, compute an estimate of matrix P. COmpare your results with the actual matrix.

Next, use $\gamma_n(j)$ to compute estimates of the Gaussian parameters. Construct a recursion that includes all elements to produce an estimate of the hidden states.

7 The Viterbi Algorithm

It is possible to construct a procedure to improve the posteriors of the hidden states usning the so called Viterbi algorithm. Explain this procedure and try an implementation using the recursion seen in the theory.