INF519 Machine Learning 2: Homework 4
Hidden Markov Model (HMM)

Kyota Lannelongue – Hugo Cambier – Thomas Charuel

Hidden Markov Model is used to reason over time or space, and more precisely reason about a sequence of observation.

In this experience, we consider a set of visible output x and a set of state z generating those output. We set a as the transition probability between states and b as the emission probability. We try to get the most probable sequence of z that generated the outputs x.

We have the transition probabilities and the emission probabilities initialized. This will help us compute the probability that the model defined by those two probabilities has generated the sequence of visible state.

```
[[ 1.   0.   0.   0. ]
 [ 0.2  0.3  0.1  0.4]
 [ 0.2  0.5  0.2  0.1]
 [ 0.7  0.1  0.1  0.1]]
```

Figure 1 Transition probability matrix

```
[[ 1.   0.   0.   0.   0. ]
 [ 0.   0.3  0.4  0.1  0.2]
 [ 0.   0.1  0.1  0.7  0.1]
 [ 0.   0.5  0.2  0.1  0.2]]
```

Figure 2 Emission probability matrix

Our visible states or outputs is as follows: $x = \{x_1, x_3, x_2, x_0\}$

By using the HMM Forward algorithm, we compute probability of the model generating the sequence of output.

```
alpha =
[[  0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
    1.82180000e-03]
 [  1.00000000e+00   9.00000000e-02   5.20000000e-03   5.19200000e-03
    0.00000000e+00]
 [  0.00000000e+00   1.00000000e-02   2.17000000e-02   5.43000000e-04
    0.00000000e+00]
 [  0.00000000e+00   2.00000000e-01   5.70000000e-03   9.64000000e-04
    0.00000000e+00]]
Probamodel forward 0.0018218
```

Figure 3 The probabilities of each step, ending with the probability of the model

We see that the model has a 0.0018218 probability of generating the model. This is very low. We can conclude that the model is not perfect. To update the model, we update the transmission and emission probabilities that define this very model by using the Baum-Welsh algorithm. We will use both forward and backward algorithm to generate the transition probabilities from $z_i(t-1)$ to $z_j(t)$

```
Backward
[[  0.00000000e+00   0.00000000e+00   0.00000000e+00   1.00000000e+00
    1.00000000e+00]
 [  1.82180000e-03   7.48000000e-03   8.20000000e-02   2.00000000e-01
    0.00000000e+00]
 [  1.62720000e-03   1.24600000e-02   5.80000000e-02   2.00000000e-01
    0.00000000e+00]
 [  6.05000000e-04   5.12000000e-03   2.40000000e-02   7.00000000e-01
    0.00000000e+00]]
```

Figure 4 Result of the HMM Backward algorithm

Using the newly generated transition probability, we update the transitions and emissions probabilities of the model.

```
a=
[[ 1.          0.          0.          0.        ]
 [ 0.26223547  0.25743724  0.12637002  0.35395727]
 [ 0.07279796  0.60933101  0.11261563  0.2052554 ]
 [ 0.36761822  0.1141861   0.44857267  0.06962301]]
```

Figure 5 Updated transition probability

```
b=
[[  1.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
    0.00000000e+00]
 [  0.00000000e+00   7.60236376e+01   1.01475277e+02   2.53614829e+01
    5.06776100e+01]
 [  0.00000000e+00   9.99594804e-02   9.98510422e-02   7.00214021e-01
    9.99754566e-02]
 [  0.00000000e+00   4.99757445e-01   1.99633553e-01   9.99533257e-02
    2.00655676e-01]]
```

Figure 6 Updated emission probability of the model

We iterate the algorithm until satisfying convergence of the model. But in our case we see that the model start to get big values.

```
alpha =
[[  0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
    9.14807026e+02]
 [  1.00000000e+00   1.95713158e+01   1.28488406e+02   3.47518631e+03
    0.00000000e+00]
 [  0.00000000e+00   1.26318814e-02   1.78834616e+00   1.67247961e+00
    0.00000000e+00]
 [  0.00000000e+00   1.76892781e-01   6.93907780e-01   9.16213903e+00
    0.00000000e+00]]
Probamodel forward 914.807026148
```

Figure 7 newly generated alpha

We see that alpha is getting very big values, throwing out our model. By fixing the implementation of alpha the algorithm should work.

```
a=
[[  1.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00]
 [  2.51827827e-01   7.37267699e-01   8.17538972e-03   2.72908440e-03]
 [  3.99100701e-03   9.94522141e-01   5.77931239e-04   9.08920975e-04]
 [  3.39839349e-01   5.68208285e-01   9.12161853e-02   7.36180013e-04]]
b=
[[ 1.          0.          0.          0.          0.        ]
 [ 0.          0.27283574  0.28083327  0.25119544  0.26519671]
 [ 0.          0.00703696  0.00316918  0.94347214  0.04632171]
 [ 0.          0.35449586  0.01829643  0.20293419  0.42427352]]
Iteration: 2
```