Kungliga Tekniska högskolan

DD2380
Artificial Intellignece

Questions' answer

Students:
Antoine Missue
Yohan Pellerin

December 5, 2023

1 Question 1

In this problem, we can calculate the transition probability matrix, $A = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$, the observation probability matrix $B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}$ and the initial probability vector $\pi = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}$.

2 Question 2

We have the following matrix: $current\_state\_distribution = \begin{bmatrix} p_1 & p_2 \end{bmatrix}$.
We can now calculate the distribution of the next state:
$next\_state\_distribution = current\_state\_distribution * A = \begin{bmatrix} p_1 & p_2 \end{bmatrix} \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}$, Since $p_1 + p_2 = 1$.

3 Question 3

If we multiply the result with the observation matrix, we get the distribution of the next state. We get:
$next\_observation\_distribution = next\_state\_distribution * B = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix} = \begin{bmatrix} 0.7 & 0.3 \end{bmatrix}$.

4 Question 4

The future state depends only on the current state and not on the events that preceded it. That's why we can substitute $O_{1:t} = o_{1:t}$ with $O_t = o_t$, because observation O_t is independent from $O_{1:t-1}$ when we condition on the X_t state.

5 Question 5

For the matrix δ, for each time $t \in [1, T]$ and for each state $i \in [1, N]$, we have a value $\delta_t(i)$. Therefore δ has $T * N$ values.
For the matrix δ_{idx}, we have the same case because we store at each time $t \in [1, T]$ and each state $i \in [1, N]$ the index of the max value. So δ_{idx} also stores $T * N$ values.

6 Question 6

When we use conditional probability, $p(X_t = x_i, X_{t+1} = x_j | O_{1:T} = o_{1:T}) = \frac{p(X_t = x_i, X_{t+1} = x_j, O_{1:T} = o_{1:T})}{p(O_{1:T} = o_{1:T})}$

Then as, $p(O_{1:T} = o_{1:T}) = \sum_{j=1}^{N} \alpha_T(j)$ we need to divide by the sum over the final α values for the di-gamma function.

If we continue to derive the expression, we will find that $p(X_t = x_i, X_{t+1} = x_j, O_{1:T} = o_{1:T}) = \alpha_t(i)ai, jb_j(0t + 1\beta_{t+1}(j)$

7 Question 7

If we use exactly the same algorithm as we used to solve the hmm3 problem, we get the following result (if we increase the maximum number of iteration):

- The algorithm stop after 956 iterations for 1000 emissions.
- The algorithm stop after 1524 iterations for 10000 emissions.

But the fact is that the number of observations (or iterations) the algorithm need to converge highly depends on the definition of convergence we used. In this case, we used the definition used by Mark Stamp in his solution to solve the problem. His definition remains on the fact that $P(O|\lambda)$ is increasing at each iteration.
If we now look at some other definition of the convergence. The convergence can be defined as the norm convergence. The value of $\|current\_A - A\|$, $\|current\_B - B\|$ and $\|current\_\pi - \pi\|$ should be really small (their limit should be equal to 0 but to compute it we can stop the convergence when they are small enough).
If we execute the code with this definition and we set a minimum threshold to 0.1, we get the following result:

- The algorithm doesn't converge after 10000 iterations with 1000 emissions.
- The algorithm doesn't converge after 10000 iterations with 10000 emissions.

The problem is probably the fact that the threshold we used was too small.

8 Question 8

If we keep the same entries as we had in the question 7, we got the following results:

- For 1000 emissions and 10000 iterations: $A = \begin{bmatrix} 0.686459 & 0.111407 & 0.302135 \\ 0.097944 & 0.806723 & 0.095333 \\ 0.20102 & 0.293615 & 0.505365 \end{bmatrix}$ and $B = \begin{bmatrix} 0.697333 & 0.231655 & 0.071011 & 0.0 \\ 0.067709 & 0.417194 & 0.281188 & 0.233909 \\ 0.0 & 0.0 & 0.354672 & 0.645328 \end{bmatrix}$
- For 10000 emissions and 10000 iterations: $A = \begin{bmatrix} 0.694155 & 0.038791 & 0.267054 \\ 0.117449 & 0.738351 & 0.144201 \\ 0.153212 & 0.248767 & 0.598021 \end{bmatrix}$ and $B = \begin{bmatrix} 0.709688 & 0.186623 & 0.103688 & 0.0 \\ 0.099083 & 0.42583 & 0.314626 & 0.16046 \\ 0.34633 & 0.176861 & 0.189405 & 0.599101 \end{bmatrix}$

Therefore we can calculate the distance between those matrices and the expected results with the norm of the matrix. If we do so, we obtain:

- For 1000 emissions and 10000 iterations: $\|A - expected\_A\| = 0.082547$, $\|B - expected\_B\| = 0.203971$.
- For 10000 emissions and 10000 iterations: $\|A - expected\_A\| = 0.144677$, $\|B - expected\_B\| = 0.372662$.

It seems that we get closer with less emissions.

The distance between these matrices doesn't reflect directly the distance between the approximation and the real model. To measure the distance between the approximation and the real model we could use a metric distance such as Kullback-Leibler (KL) divergence.
For example, to evaluate the emission matrice the KL divergence is defined as :
$D_{KL}(B_{expected}||B) = \sum_{observations} B_{expected}(observation|state)log(\frac{B_{expected}(observation|state)}{B(observation|state)})$
And for the transition matrix :
$D_{\mathrm{KL}}(A_{expected}\|A) = \sum_{i,j} A_{expected}(s_i \to s_j)_1 \log\left(\frac{A_{expected}(s_i \to s_j)}{A(s_i \to s_j)}\right)$
Then, for 1000 emissions and 10000 iterations, we get 0.027 for the transition matrix and 2.032 for the emission matrix.
Then, for 10000 emissions and 10000 iterations, we get 0.034 for the transition matrix and 0.583 for the emission matrix.

9 Question 9

In this part, we changed the code to get 2 hidden states and 4 hidden states.
In the case with 2 hidden states, when we used 10000 emissions, it took only 315 iterations to converge, which is really fast. This could mean that using 2 hidden states is too simplistic compare to the real model and therefore might under fit.
In the case with 4 hidden states and 10000 emissions, it did not converge. That might means that the model captured some noise in the data and interpreted it as significant things. Therefore the model might over fit.
The best case is definitely when we use the same number of states than the number of states that are defined in the problem. In this particular scenario, 3 hidden states is the best number of hidden states because it will not under fit nor over fit.We can also think about changing the number of observations but the problem is that it would not correspond to the possible observations in the emissions since we have exactly 4 possible observations in the emissions.
When we don't know the optimal setting in advance, as we did, we can evaluate the likelihood ($P(O\|\lambda)$) and take the model with the best likelihood. with the same number of iterations. If we have more data, a more complex model might with more hidden states might be useful because it has more information to learn from. In opposition when we have less data fewer hidden states is preferable to avoid overfitting.

10 Question 10

When we initialize our Baum-Welch algorithm with a uniform distribution, the algorithm converge very quickly but the result is very far from the real matrix. The same result is observed with a diagonal matrix.
When we initialize our Baum-Welch algorithm with a matrix that are close to the solution, we get a good result and only after 376 iterations