**Computational Cognitive Science (CS786), Spring 2019**
**Indian Institute of Technology Kanpur**
**Homework Assignment Number 4**

*Student Name:* Anurag Pandey
*Roll Number:* 160140
*Date:* May 2, 2019

# QUESTION 1

The first problem assumes that the world can be represented as a set of draws from N (here N = 5) independent Gaussians, and that the means of these Gaussians evolve over time linearly, while their variances remian fixed.

We have a randomly fixed presentation schedule. We assumes their is a constant change in the mean of these Gaussians given by delta.

Any item, we wish to encode in the memory, we do it in association with the state of the world at that time. So for the $m^{th}$ item, we encode it in the memory by the state of the world at the time this item was learnt. Also we store this item in its encoding as in memory we encode the concept and cue together.

At Test Time, we calculate the strength of association( SOA ) of the current world with the m items we have stored in our memory. Their SOA is given by their dot product, normalised by a constant. Using this SOA values calculated we get the item retrieved at each unit of the test time. Finally success is given by the number of distinct items we could retrieve in the Test Time period.

There is a possibility that none of the items are retrieved at a unit time during the testing phase. To handle this, corresponding to all such unit times, we enter 0 in the *out* array. At the end of the test time , we add an extra 0 in the *out* array. Finally we subtract one from the number of distinct entries in the *out* array.

**Computational Cognitive Science (CS786), Spring 2019**
**Indian Institute of Technology Kanpur**
**Homework Assignment Number 4**

*Student Name:* Anurag Pandey
*Roll Number:* 160140
*Date:* May 2, 2019

# QUESTION 2

For this problem, we assume that the world contains context changes, and represent this fact by sampling the rate of drift over time (delta) in the feature means itself from a mixture of two Gaussians, one with a small mean to denote small changes, and one with a large mean to denote large changes. Basically this is to incorporate the fact that the change in context can be slow at times while can be rapid at other times.

We implemented this in our model. Now one such schedule which ensures a very good retrieval rate is pushing all the items in the end. This will ensure we have high SOA with all the items in the memory and thus a majority of them are retrieved. High SOA implies high dot product between the state of the world at present to that at time of their encoding. A rough estimate of high SOA comes from the fact that if the items are present at the end of the schedule, the mean of gaussians from which the encoding of items have been sampled will be close to the mean in the current world than if the items where present at the beginning of schedule or anywhere else.

So putting all the items at the end of the schedule list gives us good retrieval rate. But it increases the encoding load. Just like memorizing the entire syallabus a night before exam is very good considering retrieval rate but its difficult and hence unlikely one can cover the entire syllabus. *( high encoding load )*.

So, we have a trade off between minimising encoding load ( *schedule load*) and achieving high retrieval rates. To minimise the encoding load we need to maximise the median of array of difference in encoding times of consecutive items. At the same time to get good retrieval rates we need to put a majority of items towards the end of schedule.

Say the maximum median is $x$. Now we put half of the items at the end of the list say at 496 , 497 , 498 , 499 , 500 respectively. Now we place the other half items such that they are equi-spaced among themselves i,e at $496 - x$ , $496 - 2x$ and so on. Satisfying all the constraints we find that the optimal value of $x = 99$.

Thus, the optimal schedule is when the items are encoded at the following times : 1 , 100 , 199 ,298 , 397 , 496 , 497 , 498 , 499 , 500. This gives the value of encoding load = 5.0505.

**PS:** This is assuming the retrieving agent knows the parameters of the model that will generate his world contexts.

**Computational Cognitive Science (CS786), Spring 2019**
**Indian Institute of Technology Kanpur**
**Homework Assignment Number 4**

*Student Name:* Anurag Pandey
*Roll Number:* 160140
*Date:* May 2, 2019

# QUESTION 3

This problem is same as the last one, but this time we are assuming that the retrieving agent does not know the parameters of the world generating model. He just knows the fact that the rate of drift is sampled from a bimodal distribution.

We can use Expectation Maximization( *EM* ) algorithms to learn the parameter of the bimodal distribution, which the retrieving agent can use during the test time.

Applying EM algorithms on the delta, we learn the mean and variances of the distributions being mixed and the probability ratios in which they are mixed. During the test time, delta is sampled from this learnt distribution. We get quite good retrieval results using our EM algorithm method, with an average success greater than equal to 7.

**References** : Expectation Maximisation