CSE/ECE-471: Statistical Methods in AI **Spring 2020**

ML for Sequential Data (HMM)

Prepared by: Abdul Wazeed (2019201054), Yash Upadhyay (2019201098) and R V sricharan priyatham (2019202001)

Consider a domain of problems where the data points are not independent of each other and there is an ordering between these data points (i.e.) nearby x and y values are likely to be related to each other. One example for such kind of problem is Auto completion where the next character suggestion depends on what sequence of characters you have type so far. These kind of problems can be dealt by building a model based on classic machine learning framework, but such model will have poor accuracy as it fails to capture the sequential pattern present in the data. A model which can capture such sequential pattern can give a better prediction accuracy.

In this note, we will discuss about sequential data and framework,sequential task,auto regressive models,Face - Based Modelling of Underlying Mood,Sequential Processes,Relationship between observed process and the hidden states,Modelling observation probabilities,Identifying states,Summary of HMM parameters and The Three Problems of HMM in the following sections.

1 Sequential Data and Framework

To better understand why we need to consider sequential data as a different paradigm and learn a new framework to deal with sequential data, we need to know the issues with classical machine learning framework.

a) A classical machine learning framework deals with fixed length input vectors. So,it will not work for variable length input vectors.

b) Suppose, you are performing the task of predicting the next word and your input is:

" In France,I had great time and I learnt some ____ language".

The answer to this is French which your classifier can predict only if it had learned that there is word "France" in your input. Classical framework can't preserve such dependencies.

c) You are performing the task of sentiment analysis where you have to predict the mood based on the given text data. Let's say you have decided to build a bag of words to create your features. If two of your data samples are:

1. "The food is good,not bad at all".
2. "The food is bad,not good at all".

For both of these data samples, the bag of words representation is same but the data sample 1 indicates good mood, while data sample 2 indicates bad mood. By just changing the ordering of the word the meaning of the data has changed a lot. Classical framework can't maintain this ordering in the data.

Considering these issues, we can understand that the data we are working on is of variable length, it has dependencies between its points and only a particular ordering can give it proper meaning. Such data is your Sequential data.

Now, in mathematical perspective, your sequential data is in the form:

$$X = \langle\ (x_1, y_1), (x_2, y_2), ...(x_n, y_n)\ \rangle$$

where each of these points (x_i, y_i) are dependent on each other and X is a single data sample in your dataset. Hence, if your= try to characterize your data using a distribution, then the probability of an individual element in this data sample is :

$$P(X) = P(x_1, x_2, .., x_n)$$

2 Sequential Task

Given some sequential data, we need to know what kind of tasks we are interested in performing with this data. Sequential tasks can be classified into two categories based on the domain of output we desire, We shall understand these categories with examples.

1. Output belongs to some class,value or some sequence which is from another domain-

   * Target is Class :

   ex- Sentiment analysis, Output is some kind of label like happy / sad based on the input sequence.

   * Target is Value:

   ex- Output is Predicting Virality score of a virus based on its genetic sequence.

   * Target is another sequence :

   ex- Text translation from one language to another like English to Hindi. (Here the domain of output is hindi text and domain of input is english text)

2. Output belongs to same domain as input-

   This is also called as self-supervision as we are trying to predict values which belong to the same domain as input sequence. Examples- Predicting the future price of a particular stock based on history of prices of that stock. Both the input sequence and output sequence belong to the same domain, that is, prices of a particular stock

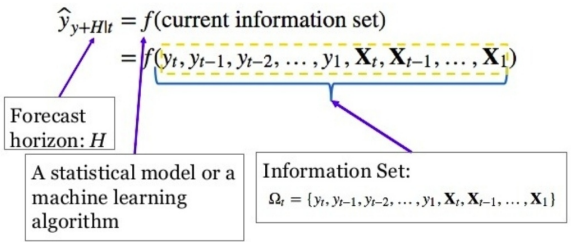
3 Auto Regression Model

Before we discuss about machine learning models for sequential data, let us see what kind of models does statistics has to offer.

Autoregressive (AR) models are statistical models which predict future values based on past values. Autoregressive models operate under the premise that past values have an effect on current values, which makes the statistical technique popular for analyzing nature, economics, and other processes that vary over time.

An AR model predicts future behavior based on past behavior. It's used for forecasting when there is some correlation between values in a time series and the values that precede and succeed them. You only use past data to model the behavior. The process is basically a linear regression of the data in the current series against one or more past values in the same series.

Figure given below shows a mathematical way to represent autoregressive models. Here forecast horizon H represents the window size of your output.



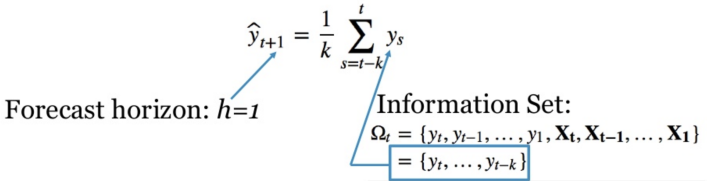
The AR process is an example of a stochastic process, which have degrees of uncertainty or randomness built in. The randomness means that you might be able to predict future trends pretty well with past data, but you're never going to get 100 percent accuracy. Usually, the process gets "close enough" for it to be useful in most scenarios. Some of the autogressive models are : AR(1),AR(P), ARMA,etc.

1) AR(1)-

$$\hat{y}_{t+1} = y_t$$

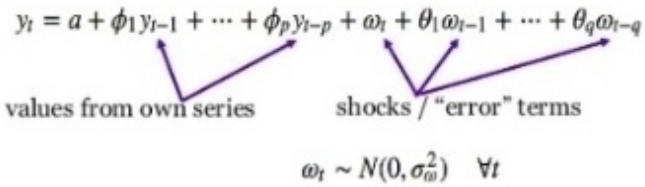
This is the simplest among the AR models as it states that the current value is based on the immediately preceding value.

2) AR(P)-



The AR(P) model is one in which the current value is based on the previous P values.

3)ARMA-

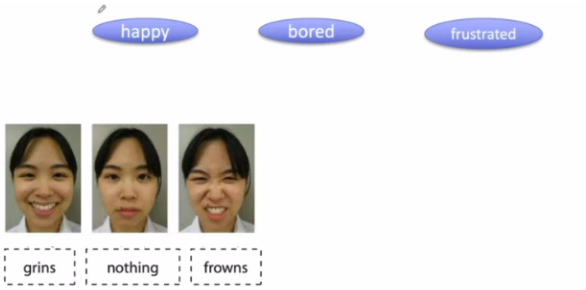


In these models, in order to predict a particular value,you can include certain modelling assumptions in form of errors that are present in the model along with the input values. These errors in modeling can be represented in form of a Gaussian distribution denoted by $N(0, \sigma_\omega^2)$

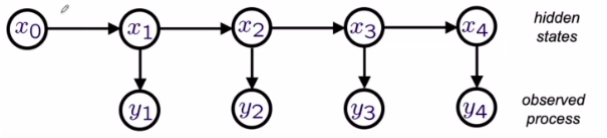
There are issues with auto regression models. Since autoregressive models base their predictions only on past information, they implicitly assume that the fundamental forces that influenced the past values will not change over time. This can lead to surprising and inaccurate predictions if the underlying forces in question are in fact changing, such as in case of predicting the weather where we assumed temperature during April will be similar to temperature during March but actually it is much hotter during April than in March. This happened as we didn't account the seasonality change from spring to summer.

4 Face - Based Modelling of Underlying Mood

Suppose we want a system in which our model uses facial expression. Like student is attending class and we want know what part of lecture student found happy, bored or frustrated , on the basis of what are its facial expression during the entire session. Considering a case here, a person can be in three moods



happy, bored and frustrated and corresponding to these facial expression are grins, nothing and frowns. Here we have facial expression as our data, we don't know about mood. So now, if we want to model the underlying mood, we might have a model like this, Here X=moods and Y=facial expression(input data).



Obviously here our X is invisible and Y is visible. So let us assume that we have this process going on, so you start out in with some mood and at timestamp 1 you are happy and because you are happy that causes your face to look in a particular manner because of which some system says that you are grinning. So this is our model. So in contrast to what we do for Gaussian mixture models in Gaussian mixture models. If you remember, we said that you can have a model for explaining how your data came, and that model was like I pick a cluster first. So there are many clusters. I pick a cluster and the manner in which I picked this cluster is based on some component probability. So if there are three clusters, so maybe this is the most likely cluster, so this cluster will be picked more frequently on average. So I pick one cluster and associated with this cluster is a Gaussian distribution. So when I sample from that particular Gaussian distribution ,that causes a particular data sample to be generated. Then, go ahead and distribution has a certain mean, certain variance, and that causes a sample to be distributed. So this is my explanatory model and my generative model for data .In the same way, what I'm doing right now is I have a model which I'm using to characterize this entire process, so I would like to find out what the hidden states are ?. So that is mine underlying objective. So my model is, I start out in a particular state when there was no data and then I transition to a state called X1 and because my mood is happy, it results in generation of this particular facial expression which is recognized as a grins. In the next time step, just say, I'm still happy and after time T = 200 milliseconds, I am still grinning and so on. So in our model corresponding to each X we have Y. So now I'm using the language of a similar to what we use for GM's. What is the probability that my underlying mood is a frowning, given that the expression is grins, what is that probability that my underlying state is board given that my expression is nothing. So in order to solve the problem in this manner we use the machinery of hidden Markov models. We make certain assumptions and that's what we're going to look at later. So given this video of student attending class , I want to have a model which will give me these statistical model which will give me, for a given video, what is the underlying set of mood states. It gives me some distribution of mood states. So till now we now Y's and we don't know X's and parameter of temporal model.

5 Sequential Processes

Sequential processing refers to the process of integrating and understanding stimuli in a particular, serial order. Both the perception of stimuli in sequence and the subsequent production of information in a

specific arrangement fall under successive processing. Consider a system which can occupy one out N states, like in our case N=3 and possible states are happy, bored and frustrated.

$$x_t \in \{1, 2, \ldots, N\} \rightarrow \text{state at time } t$$

Here Xt represent mood at time t. We want to model this as a random process because we are interested in stochastic systems. Here both x and y are random. Specifically we make a simplifying assumption ,that is Markov Assumption, which states that, the next state depends only on the current state,

$$p(x_{t+1} \mid x_0, \ldots, x_t) = p(x_{t+1} \mid x_t)$$

So here assumption states that if we need to find distribution at per timestamp t+1 given the values of random variables x from timestamp [0 to t], where x are moods states , we only need x at current timestamp ,that is , it is affected only by x*t*. Now as per our example, x can be 3 in states at any instance.

$$Q = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}$$

$$q_{ij} \triangleq p(x_{t+1} = i \mid x_t = j)$$

If you notice, there is some kind of a jumping between different states, right? So initially you could be in a bored state and you can continue to be in bored state, then to some other and so on. So there is some kind of a transition between these modes, so to capture this modeling we use transition matrix .So the transition matrix captures this particular probability(Qij here). That is, if, my current state is this j (say j=0, i.e. bored) at time t, what is the probability that in the next time step my state is going to be i. Probability of i=j here means that between timestamp t and t+1 mood remain same.

There 2 constraints here: 1. Qij is non-negative. 2. column sum is always one.

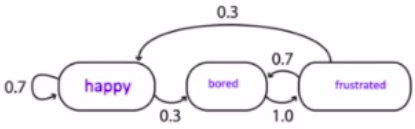
For example, this might be the state transition matrix for our data, with s1 as happy, s2 bored and s3 frustrated.

	s_1	s_2	s_3
s_1	0.5	0.1	0.7
s_2	0.3	0.5	0.2
s_3	0.2	0.4	0.1

But in reality we have to learn this matrix. So this is something that you need to learn. You do not get this given to you. So from data you somehow have to learn. What is the state transition matrix possible.

Another example could be like the one given below.

The probability that if you are in a happy state will remain in the Happy State is 0.7. If you're happy, the probability that your mood will change to bored is on average 0.3 probability that the mode will change to bored. So this is to remember that these are stochastic processes. So our matrix will be based on what is going to be my probability distribution over the next set of moods that are going to be there. That is what we are capturing. So the distribution over the set of modes for my next time step, assuming that my current time step is happy is given by this particular set of probabilities. Similarly, when I'm bored, I have another set of probabilities, right? So this is the set of probabilities that come into picture when my current state is bored. Similarly for frustrated. So you can see that this transition matrix is capturing the dynamics of what is happening. This is very valuable information.



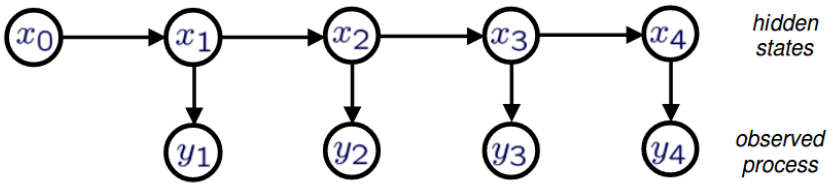
$$q_{ij} \triangleq p(x_{t+1} = i \mid x_t = j)$$

$$Q = \begin{bmatrix} 0.7 & 0.0 & 0.3 \\ 0.3 & 0.0 & 0.4 \\ 0.0 & 1.0 & 0.6 \end{bmatrix}$$

Note: here Q13 = 0 not 0.3.

6 Relationship between observed process and the hidden states

To summarize the above what we done so far i.e, defining hidden states, learning transition matrix etc.., comes under transition modelling. From this section to subsequent sections we will see how to develop a observation model of HMM. Let's consider four hidden states and four observed states as shown in the figure. Notice that it is not necessarily a one on one connection between the hidden states and observed



states strictly as shown in the figure. In this figure all the x's contribute to hidden states and all the y's contribute to observed processes.

6.1 Markov Assumption II

It states that earlier observations provide no information about the current or future observations given current hidden state i.e, current or future observations are independent of the previous observations. The same can be expressed in probabilistic terms as follows.

$$p(y_t, y_{t+1}, \ldots \mid x_t, y_{t-1}, y_{t-2}, \ldots) = p(y_t, y_{t+1}, \ldots \mid x_t)$$

where y_t denotes the current observed state, x_t denotes the current hidden state. y_{t+1},y_{t+2}... are the future observed states and and y_{t-1},y_{t-2}... are the previous value of observed states.

6.2 Probabilistic analysis of a sequence of hidden states

Based on two Markov's assumptions stated above with little effort we can derive the probability for a sequence of hidden states to occur and the result is as follows. Observe that the below rhs is a partital

result and must be summed over all the values of y_i (i.e observed process) to get the joint probability of hidden states.

$$p(x_0, x_1, \ldots, x_T) = p(x_0) \prod_{t=1}^{T} p(x_t \mid x_{t-1}) p(y_t \mid x_t)$$

7 Modelling observation probabilities

We can now associate each hidden state with different observation distribution notice that here hidden states are discrete but observation distribution can be both discrete and continuous. Observation densities are typically chosen to encode domain knowledge

7.1 Discrete Observation Distribution

If we have a discrete(finite) observation processes say M each hidden state will maintain a column of size M corresponding to the probability of transition to each state. For example in the figure shown above

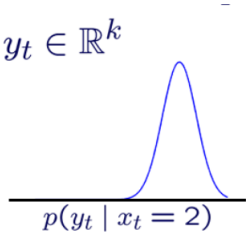
$$y_t \in \{1, 2, \ldots, M\}$$

$$p(y_t \mid x_t = 2) = \begin{bmatrix} 0.2 \\ 0.2 \\ 0.1 \\ 0.5 \end{bmatrix}$$

at $x_t = 2$ we have M=4 transitions to observed states possible

7.2 Continuous Observation Distribution

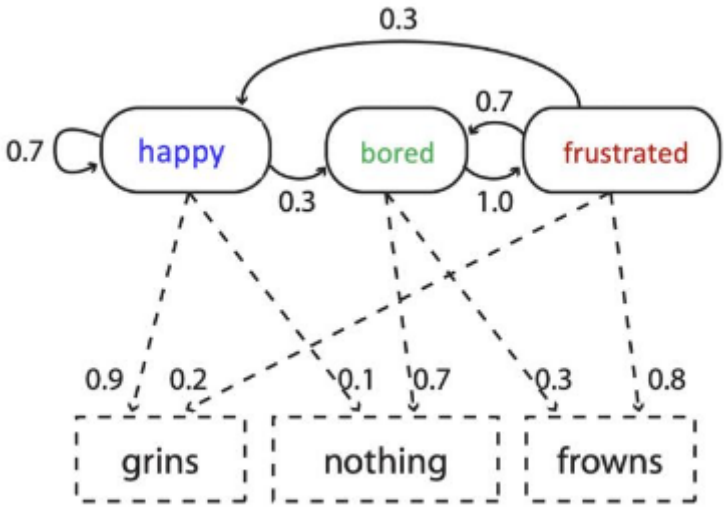
If we assume observation distribution to follow some continuous distribution (such as gaussian) at each hidden state we need to store parameters such as mean of the distribution, variance of distribution etc., For example in the figure shown above at $x_t = 2$ we the observation model follows a guassian distribution



with some fixed mean and variance.

7.3 Representation of observed probabilities

After the observation probabilities were modelled the updated transition diagram of the face based modelling of underlying mood. looks as follows.



Along with the probabilities that are explained earlier notice that there are new transitions(represented with dotted lines) from hidden states to observed states for example a dotted arrow line from happy to grins states that for the hidden state corresponds to happiness the observed expression will be grins with a probability of 0.9 similarly if the hidden state corresponds to frustrated the observed expression will be frown with a probability 0f 0.8

8 Identifying states

We decide states mostly by domain knowledge. For example If it is speech we know how many phonems are possible. If we are lucky we can get these insights from training data itself. Examples to name a few are follows.

- Analysis of a physical phenomenon
  - Dynamical models of an aircraft or robot
  - Geophysical models of climate evolution
- Discovered from training data
  - Recorded examples of spoken English
  - Historic behavior of stock prices

9 Summary of HMM parameters

The entire HMM which we have seen so far can be broadly divided into three models. A fully trained model of HMM should have proper values defined for each of this entity.

- Initial state distribution
  - It involves defining Initial probability $P(x_0)$
- Transition model
  - Involves modeling of transition matrix from One Hidden state to other i.e $P(x_i/x_{i-1})$
- Observation model
  - Involves modeling of Observation distributions corresponding to each hidden state i.e $P(y_i/x_i)$

10 The Three Problems of HMM

Evaluation: How likely is a given sequence of observations, given parameters of our model ?

Decoding: What is the most likely sequence of hidden states, given the sequence of observations and parameters of the model ?

Learning: Given many sequences of observations, how do we determine the parameters of the model ?

References

[1] ML for Sequential Data, Statistical Methods in AI by Ravi Kiran S and Vineet Gandhi.

[2] https://www.statisticshowto.com/autoregressive-model/

[3] Sequential Data, Pattern recognition and Machine Learning by Christopher M. Bishop