Project Report

Statistical Methods in AI & Machine Learning
CS 6347.001

Topic: Integer Sequence Learning

Group Members:
Aditya Mahajan - axm156630
Hardik Bhadja - hxb162230
Karan Motani - kbm160230

Introduction:

- Purpose:
  - It follows a machine learning algorithm capable of guessing the next number in an integer sequence.
- Explicit statement of problem:
  - To predict the next number in sequences of integers based on the statistics of the preceding numbers in the sequence.

Methods:

- Hidden Markov Model(HMM):
  - It is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states.
  - It can be presented as the simplest dynamic Bayesian network.
  - In simpler Markov models (like a Markov chain), the state is directly visible, and therefore the state transition probabilities are the only parameters, but in HMM the state is not directly visible, but the output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens.
  - Therefore, the sequence of tokens generated by an HMM gives some information about the sequence of states.

  - A HMM consists of three types of probabilities:
    - 1. The initial hidden state probability(π)
    - 2. The transition probability from one hidden state to another(Q)
    - 3. The emission probability for a given observation given the hidden state.

  - We initialized these probabilities using a uniform distribution

- Baum–Welch algorithm:
  - It is used to find the unknown parameters of a Hidden Markov Model (HMM).
  - It's basically another version of the EM algorithm, which uses a technique called forward-backward method to calculate the probability of a hidden state given the observations.

- Steps:
    1. Assume a number of hidden states.
    2. Take a set of observations
    3. Estimate a new transition matrix, emission matrix and the initial probabilities using the Baum Welch algorithm.
    4. Repeat the above steps until the resulting matrices converge satisfactorily.
    5. Take the most probable hidden state and output the observation with the highest probability for that given hidden state.

- Overall procedure:
    - **In the first pass:** the forward–backward algorithm computes a set of forward probabilities which provide, the probability of ending up in any particular state S_i, given the first *t* observations $C_{1:t}$ in the sequence.

$$P(S_t = s_i | C_{1:t})$$

    - **In the second pass:** the algorithm computes a set of backward probabilities which provide the probability of observing the remaining observations $C_{k+1:t}$ given any starting point *k*, S_k.

$$P(C_{k+1:t} \mid S_k)$$

    - These two sets of probability distributions can then be combined to obtain the distribution over states at any specific point in time given the entire observation sequence.

$$P(S_k \mid C_{1:t}) = P(S_k \mid C_{1:k}, C_{k+1:t}) \propto P(C_{k+1:t} \mid S_k) P(C_{1:k}, S_k)$$

- Prediction of the next state S_{t+1} given the observation $C_{1:t}$ can be computed from qS_t, S_{t+1}, which is transition probability of state transition from S_t to S_j, and $P(S_t|C_{1:t})$, according to the law of total probability:

$$P(S_{t+1} = s_j | C_{1:t}) = \sum_{i=1}^{M} q_{s_i, s_j} P(S_t = s_i | C_{1:t})$$

    - where $P(S_t = s_i | C_{1:t})$ can be computed using the "filtering" approach. Specifically, we compute $P(S_t = s_i | C_{1:t})$ from Equation below:

$$P(S_t = s_i | C_{1:t}) = (\pi \mathcal{P}_{C_1} \mathcal{Q} \mathcal{P}_{C_2} \mathcal{Q} \cdots \mathcal{P}_{C_{t-1}} \mathcal{Q}_i P(C_t | S_t)) / L_t$$

    - π is the initial probability matrix, Q_i is the i^{th} column of transition matrix Q, and L_t is the likelihood of $C_{1:t}$ computed from Equation below, where P_{Ci} is the i^{th} entry from the emission matrix, probability of the observation given the state:

$$L_T = P(C_{1:T}) = \pi \mathcal{P}_{C_1} \mathcal{Q} \mathcal{P}_{C_2} \cdots \mathcal{Q} \mathcal{P}_{C_T} \mathbf{1}^\top$$

- After getting the most likely hidden state S_{t+1}, we obtain the category distribution $P(c_j | S_{t+1})$, $1 \leq j \leq N$, and predict the most likely category as follows:

$$S_{t+1} = \arg\max_{s_i \in S} P(S_{t+1} = s_i | C_{1:t})$$
$$C_{t+1} = \arg\max_{c_j \in C} P(c_j | S_{t+1})$$

Results:

- Input:
  - Dataset test.csv which contains the sequences of integers.

- Output:
  - Number of sequence for which the prediction is correct. E.g last element predicted is same as the actual number in the sequence.

- Programming Framework:
  - Language: python 2.7
  - Package: Numpy

- Experiments Outcome Chart:

# Observations	Number of Sequences	Time Estimate (seconds)	Accuracy
1	50	5	46%
2	100	8	32%
3	500	40	22.4%
4	1000	80	22.9%
5	5000	420	24.66%
6	10000	1000	24.54%

- Explanation:
  - Searching for the best number of HMM, e.g number of hidden state, along with the large number of input integer sequence takes it longer than usual to get the convergence and hence the prediction result.

Discussion:

- Source of Input Data:
  - It is imported from www.kaggle.com.
  - This dataset contains the majority of the integer sequences from the OEIS.

- Interpretations of the input data:
  - Remove the last number from the sequence, and run the algorithm to predict the last element.

- Alternative explanations:
    - This could be related with the notion of supervised learning, in which we have the next element as the last element of the sequence itself, and we learn next element of the sequence with the last element trimmed by means of the procedure explained above, and check that predicted one (e.g. next element) with the actual last element.

- Suggestions for further investigation:
    - Instead of using the uniform distribution of limited domain of integer numbers for the prediction, we could use different probability distribution e.g dirichlet distribution, for the large integer number as the domain to increase the odd of correct prediction of the next number.

Conclusions:

- Conclusion of the problem statement:
    - This way next element of the integer sequence can be predicted but, the accuracy is not up to the mark of highly efficient predictor.
    - Problem solving methodology needs strong co-relation in the statistics of the previous elements in the sequence.

- Significance of the findings:
    - The efficient implementation of this problem statement can be used in Sequential Decision Making, adaptive control, time series prediction, financial engineering, DNA sequencing, and so on.

References:

- Technical reference:
    - Kaggle. "Integer Sequence Learning." Integer Sequence Learning | Kaggle. 25 April 2017. <https://www.kaggle.com/c/integer-sequence-learning/data>
    - Jihang Ye, Zhe Zhu and Hong Cheng. "What's Your Next Move: User Activity Prediction in Location-based Social Networks." <http://www1.se.cuhk.edu.hk/~hcheng/paper/sdm2013.pdf>. 2013: 4-6
    - Stephen Tu. "Derivation of Baum-Welch Algorithm for Hidden Markov Models." Hmm-baum-welch-derivation. <https://people.eecs.berkeley.edu/~stephentu/writeups/hmm-baum-welch-derivation.pdf>
    - Wikipedia. "Baum-Welch Algorithm". Baum-Welch algorithm – Wikipedia. 25 April 2017. <https://en.wikipedia.org/wiki/Baum%E2%80%93Welch_algorithm>
    - Wikipedia. "Forward-Backward Algorithm". Forward-Backward algorithm – Wikipedia. 25 April 2017. <https://en.wikipedia.org/wiki/Forward%E2%80%93backward_algorithm>