Bayesian Decoding of Neural Spike

Li Yuhui 2013012470

Experimental Setup

All the codes are written in python 2.7 and can be run in windows 10. Useful packages are included such as numpy, scipy, cPickle and hmmlearn.

Data Source

Dataset is provided by TA, which includes 7 neural spike data through time and the one-dimensional position of rats for each.

Data Preprocessing

Here, each data for unit time (smallest interval) is considered as an example. Neural spike data constitutes the observation, and the position of rats constitutes the hidden state. For neural spike data, an m\*p matrix is constructed where m is the example number and p is the recorded neuron number. For position data, a one-dimensional sliding window is used to divide the whole continuous coordinate without overlap. The number of state can be thus decided by equation (1).

(1)

Bayesian Decoding

The Naïve Bayesian model is based on Bayes’ theorem with independence assumptions between predictors. A Naïve Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naïve Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods. Here, a Naïve Bayesian model is used to decode the state of rats which is described in equation (2).

(2)

Where, is the position at time t, is the spikes at time t.

Hidden Markov Model Decoding

A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. In this problem here, one important information is the relation between the last position and the next position of rats. Therefore, the HMM model is built to describe the problem more exclusively, the general architecture is shown in Fig. 1.

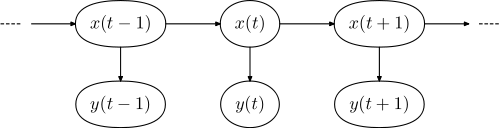


Figure 1. The Architecture of Hidden Markov Model. Here, is the position of rats (hidden state), is the neural spike of rats (observations).

The Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states – called the Viterbi path – that results in a sequence of observed events. The Algorithm is described in equations following.

(3)

(4)

Where is the probability of the most probable state sequence responsible for the first t observations that have k at its final state. The Viterbi path can be retrieved by saving back pointers that remember which state x was used in equation (4). Let be the function that returns the value of x used to compute if , or k if , Then

(5)

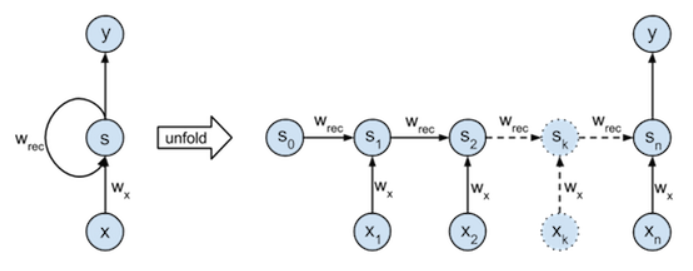
(6)

***Recurrent neural network***

A recurrent neural network (RNN) is a class of artificial neural network where connections between units form a directed cycle. This creates an internal state of the network which allows it to exhibit dynamic temporal behavior. Unlike feedforward neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs, which is typically as the equation (7) and the model is shown in Fig. 2.

(7)

Where is the state at time k, is an exogenous input at time k. and are parameters like the weights parameters in feedforward nets.



**Figure 2. The Typical Linear Model of RNN.** The left part is a graphical illustration of the recurrence relation. The right part illustrates how the network is unfolded through time over a sequence of length n.

RNN has been proved to be successful in the field of natural language processing and sequence analysis. Therefore, it should have more or less good performance in this decoding process (different from HMM, RNN is a discriminant model). Here, a naïve RNN is used to do the project.

Model Evaluation

For model evaluation, the error based on location is considered as equation (8).

(8)

Where, f(x) is the output of the model, y is the true recorded position of rats.

Parameter Setting

For Bayesian Decoding ,the only parameter is the sliding window size. Here the window size is set as {10, 20, 30, 40} as the rat speed and the feature number is considered (as p neruons can only code 2p states).

For Hidden Markov Model, the transition matrix and emission matrix is calculated as discrete function. The emission probability is considered as multinomial and the start probability is considered as equal for each window. Considering the activity of rats, the end of the coordinates has greater possibility for rats to locate, which means the start

Result

For Bayesian decoding,