Evan Coulson

Prof. Gu

Math 189Z

24 April 2020

Article Summaries

Gene finding model

Genes are easy to find in prokaryotes as we can use ORF's with a threshold to find the sequences that are most likely genes. However, it is more difficult in eukaryotes as their coding (exon) regions also contain non-coding regions (introns). we can use an HMM for segmentation and gene finding in eukaryotes. The HMM assumes that the sequences have been generated from some hidden sequence. The model works to estimate the emission matrix E and the transition matrix using a given observed sequence and hidden sequence or to estimate the most likely hidden sequence given an observed sequence. H* maximizes the probability of obtaining the maximum likelihood hidden sequence notated s. We can use the Viterbi algorithm to make the HMM tracible, which allows us to find solutions to segmentation and gene finding. The Viterbi algorithm uses dynamic programing to find its solutions, it uses the following parameters: an emission Matrix, a set of observable states, a set of hidden states, a transition matrix, and an initial probability distribution. The initial probability distribution describes the hidden states, so we know that

$$P(h_1, h_i) = p_i$$

which is the initial probability. Dynamic programming relies up recursion and a means to cache data, the Viterbi algorithm uses recursion to calculate the probability of

$$V(i, j+1) \text{ using } V(i, j),$$

the probability of hidden states most probably associated with the prefix s[1:j] and ending at $h_|$, that is cached in a table. The probability function determines the most likely transition, which is found using the pointer matrix that stores the index. Note that

$$pointer(i, j+1) = \arg max_k\{V(k, j) * T(k, j)\}.$$

We can calculate

$$V(i, j+1) = max_k\ \{V(k, j) * T(k, i) * E\left(h_|, os_{j+1}\right)\}.$$

This decides the index the maximum should be stored at. We find that the base case is then

$$V(i,1) = p(h_i) * E(h_i, os_1),$$

which fills the first column of the table used in the algorithm. The rest of the table is filled by

$$V(i,j+1) = max_k \{V(k,j) * T(k,i)\} * E(h_{|}, os_{j+1}).$$

As noted, the matrix pointer stores the decision in each cell of V. The algorithm outputs h* by tracing through the decision tracking table that has the partials stored in it. We find that the max likelihood hidden sequence (stored in the last columns of V) is

$$P(s,h^*) = max_i\, V(i,n)$$

We then find the whole sequence by building h* = $hs_1hs_2hs_3hs_4 \ldots hs_n$ after breaking them down into subproblems. Each component of h* is defined as

$$hs_n^* = \arg\big(max_i\, V(i,n)\big)$$

One issue with this algorithm is the near zero probabilities being converted to 0's due to floating point arithmetic. We solve this by converting to log space. This algorithm can then find genes with certain properties using hidden states to emit the sequence's nucleotides. In this article they demonstrated this on transmembrane proteins using two hidden states, one representing hydrophobic and the other hydrophilic). Each state was associated with the 20 amino acids. We then realize that P, T, and E aren't know so we must estimate them using unsupervised or supervised training to prepare our HMM.

HMM Stonks

This article uses an HMM because stock data is a nonstationary time series, which is not the best for linear regression. The researchers use an RSM, a variation of HMM, so that they can use a regime shifting variable as an explanatory variable. This article predicted the S&P 500 price and compared the results with the historical average return model. The researchers introduce what an HMM can find, which are: the probability of the best fit sequence given an observed sequence, the best fit sequence, and calibrates the HMM's parameters. The researchers also not that there are four main HMM algorithms: Forward, Backward, Viterbi, Baum-Welch (finds the local maximizer of the probability function). The researchers determined that a four state HMM was the best HMM and had states based of AIC, BIC, HQC, and CAIC. The prediction done by the HMM is a 3-step process that calibrates the HMM's parameters, uses a similar likelihood dataset for training data, and calculates the probability of observation until we find that our observed sequence has about the same probability of O*. The model was then evaluated using R_{os} which is a measure of the predicted and out of sample predictions with the true data and the historical average model. When $R_{os} = 0$ that indicates that the model was the same, which is our null hypothesis. If we see that $R_{os} > 0$ then we can reject the null hypothesis and state that the model out preformed the historical average model. The researchers found that HMM outperforms the historical average model in almost every case, which strongly suggests that HMM is a better model. In practice, we see that HMM outperformed HAR in all cases. This is because HMM captures change of input at a single point very well, while HAR does not. The researchers also saw that HMM beat the buy and hold strategy.

My Source

www.researchgate.net/publication/221912145_A_Non-Homogeneous_Hidden_Markov_Model_for_the_Analysis_of_Multi-Pollutant_Exceedances_Data.

A non-homogenous hidden Markov model is a flexible way to estimate a solution to the problem. The Markov model transitions based off of meteorological data. The article then discusses the effects of the environments that the monitors are in and what the levels of pollution are around it. The article also considers the effects of weather upon the measures. This article also uses a conditional multivariate distribution and calculates the marginal covariance. The HHM model occurs when the exceedances patterns are conditional independent given the latent states and the latent vector is sampled from a Markov chain. The hidden states chain can be interpreted as different regimes at which multivariate exceedances occur. The temporal persistence of each regime is determined by the latent transition probabilities of a Markov chain. The researchers expand this model to become nonhomogeneous by including a vector of meteorological covariates and use a multinomial logit model to reparametrize the time-varying transition probabilities. The gamma vector measures the regression coefficients of weather conditions on transition probabilities. The article then sets up a marginal distribution that separates temporal dependence, multivariate dependence, and non-stationary behavior. This can then measure the pollutes around the station of K generalized linear models. However, it should be noted that with Markov models it can be very difficult to assess uncertainties about the parameters this could be resolved using Bayesian information Criterion. The researchers provide results of a 3-state model. They found that exposure to pollution sources is strongly significant when pollution is present. They also found that the model moved between states that described the temperatures and pollutants that tend to be more present depending on the temperature at the time of year. The results indicated that solar radiation had a positive effect on the probability of moving particles and NO2. The graphs indicate to us that the states switch appropriately and cluster around the three pollutants. We also see that nonhomogeneous hidden Markov models work well with non-stationary time series. This model can help to predict AQI depending on the values that each of the hidden states are dependent upon and output an AQI.