02457 Non-Linear Signal Processing, Exercise 11: Sequence modeling and non-stationarity

This exercise is based on Christopher Bishop: *Machine Learning and Pattern Recognition*, sections 2.2, 2.3.5, 13.1[1].

Print and comment on the figures produced by the software as indicated below in the **Checkpoints**.

Sequence modeling

We will start our theme on sequence models in the discrete domain. Sequences of 'symbolic' or discrete time series observations are ever more important in behavioral modeling[2], digital media[3] and bio-medicine[4]. A typical scenario is that a sequence of data is observed and the aim is to detect, e.g., the corresponding behavior. Here we focus on modeling sequences and generalization in the case of non-stationarity.

Assume we have observed a sequence of discrete events $\{y_n|n = 1 : N\}$ encoding some kind of behavior B. Given the discrete observation sequences, we next would like to recognize the behavior, i.e., we use Bayesian decision theory and aim to compute the posterior probabilities. Using Bayes' rule

$$P(\mathrm{B}|\{y_n\}) = \frac{P(\{y_n\}|\mathrm{B})P(\mathrm{B})}{P(\{y_n\})}. \tag{1}$$

Hence, to make a decision we need to compute the probabilities $P(\{y_n\}|\mathrm{B})$ of the entire sequence given each behavior, i.e., the likelihood function for models B.

Markov sequence model

Here we will investigate a simple Markov chain model of such discrete sequences. In the behavioral case, each behavior would be associated with a specific Markov model.

Let y_n be a sequence of N symbols with K states. Let the *transition matrix* $a_{j,j'} = P(y_{n+1} = j'|y_n = j)$ be the probability of jumping from j to j'. To ensure that we always jump to some state, the matrix $a_{j,j'}$ must satisfy $\sum_{j'} a_{j,j'} = 1$.

Given an observation sequence the Markov model's transition matrix a can be estimated by maximum likelihood

[1]References/footnotes in this exercise are for further reading and are *not* part of the curriculum.

[2]See e.g.: Hara, K., Omori, T. and Ueno, R., Detection of unusual human behavior in intelligent house. In Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing 2002 pp. 697-706 IEEE.

[3]See e.g.: Lu, L., Dunham, M. and Meng, Y., Mining significant usage patterns from clickstream data. In International Workshop on Knowledge Discovery on the Web 2005 pp. 1-17. Springer Berlin Heidelberg.

[4]See e.g.: Jackson, C.H., Sharples, L.D., Thompson, S.G., Duffy, S.W. and Couto, E., Multistate Markov models for disease progression with classification error. Journal of the Royal Statistical Society: Series D (The Statistician), 52(2), 2003 pp. 193-209.

$$
\begin{aligned}
P(\{y_n\}|a) &= P(y_1)\prod_{n=2}^{N} P(y_n|y_{n-1},a) \\
&= P(y_1)\prod_{j,j'}(a_{j,j'})^{\mathrm{n_{j,j'}}}
\end{aligned}
$$

where the $n_{j,j'}$'s are the occurrences of the transition, and $P(y_1)$ is the probability of starting in state y_1. Since we have assumed that we only have a single sequence to learn from we will let this be estimated as $P(y_1) = 1/K$. Using softmax or Langrange multipliers to ensure the normalization condition $\sum_{j'} a_{j,j'} = 1$ we obtain the simple solution,

$$
\widehat{a}_{j,j'} = \frac{n_{j,j'}}{\sum_{j''} n_{j,j''}}
$$

A maximum posterior estimate (MAP) with a Dirichlet prior on the rows $a_{j,:} \sim \mathcal{D}(a_{j,:}|\alpha_{j,:})$ provides the maximum posterior estimate

$$
\widehat{a}_{j,j'} = \frac{n_{j,j'} + \alpha_{j,j'} - 1}{\sum_{j''} n_{j,j''} + \alpha_{j,j''} - 1}
$$

Checkpoint 11.1

Use the matlab script main11a.m to create a random transition matrix. This matrix will be used as a *teacher* to create training and test sequences.

First, we analyze the properties of Markov transition matrices. Let consider a large ensemble of parallel chains starting from the same initial state $y_1 = 1$. The n'th state of the q'th ensemble member is denoted $y_n^{(q)}$. At any given time n we can consider the distribution of states within the ensemble. Let $P_n(j)$ be the probability that ensemble members are in state j at time n. Establish an argument for the temporal evolution of the ensemble distribution, the so-called *Master equation*

$$
P_{n+1}(j') = \sum_{j=1}^{K} P_n(j) a_{j,j'}
$$

or in matrix notation

$$
P_{n+1} = P_n a
$$

what are the dimensions of vector P and matrix a?

Under mild conditions (e.g., all elements of a are positive)[5] the ensemble dynamics will converge to a fixed point distribution, called the *stationary distribution* which satisfies,

$$
P_*(j') = \sum_{j=1}^{K} P_*(j) a_{j,j'}.
$$

[5]Pillai, S.U., Suel, T. and Cha, S., The Perron-Frobenius theorem: some of its applications. IEEE Signal Processing Magazine, 22(2), pp. 62-75 2005.

This is an eigenvalue problem (explain!)

$$P_* \;=\; P_* a.$$

Under the same conditions on a the long term distribution of states attained by an individual chain will also be distributed as $P_*(j)$. Investigate the stationary distribution for the transition matrix a and explain how the program estimates it. Use the *teacher* transition matrix to create increasing length sequences, and observe how the histogram of the observed sequences converge to the stationary distribution. Explain function getint.m.

Now we turn to learning by maximum likelihood (i.e. $\alpha_{j,j'} = 1$). Verify the maximum likelihood estimate of the transition matrix (either analytically or numerically). Use Matlab script (main11a.m) to generate increasing length sequences. Train a *student* transition matrix on these sequences and show that the relative error of the student matrix converges to zero, hence, the student matrix converges to the teacher matrix for large training sets.

Non-stationary Markov sequence models

In a non-stationary environment the assumption that the transition matrix of a given process is constant in time may be too restrictive. To relax this assumption we can make the maximum likelihood estimation process dynamic. A simple scheme is to let the estimator operate on a window basis. We consider a streaming situation where we receive data sequentially and train on overlapping windows of length W, with the k'th window starting at skip $* k$: The first window contains the data $[y_1, ...y_W$, the second $[y_{1+\text{skip}}, ..., y_{\text{skip}+W}]$ and so forth. By testing the model on a windows of size W_{test} before we update the transition matrix we may obtain a running estimate of the sequence model's test performance.

Checkpoint 11.2

We design a simple "switching" non-stationarity with two Markov models taking turns. Model 1 generates the states in the intervals $[1, N_1]$ and $[N_1 + N_2 + 1, 2N_1 + N_2]$, while Model 2 generates the states in the interval $[N_1 + 1, N_1 + N_2]$.

We estimate Markov models using the MAP estimator for overlapping windows using a set of different window sizes. Compare the likelihood of the transition matrix computed on the training data and the likelihoods of the two "true" models.

We evaluate the estimated Markov models by their L_1 distances to the true model and using the log-likelihood on test data. Discuss the impact of the window size and the prior α on the estimated Markov model and these generalization performance estimates. Which window size would you recommend?

Lars Kai Hansen, November 2016.