LEARNING ALGORITHMS AND SYSTEMS LABORATORY
FACULTE DES SCIENCES ET TECHNIQUES DE L'INGENIEUR
SCHOOL OF ENGINEERING / SWISS INSTITUTE OF TECHNOLOGY

Lecturer:
Prof. ***Aude Billard***
Office: *ME A3 464*
direct: *+41-21-693 54 64*
secretary:+41-21-693 09 39
fax: *+41-21-693 78 50*
E-mail: aude.billard@epfl.ch
Web: http://lasa.epfl.ch/

Assistant:
Dr. **Basilio Noris**
Office: *ME A3 395*
direct: +41-21-693 7824
E-mail: basilio.noris@epfl.ch

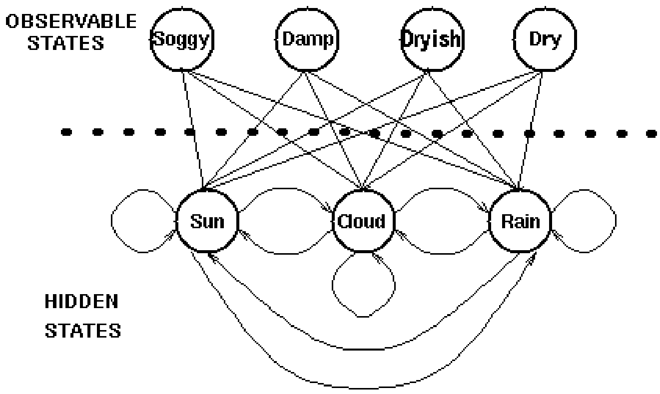
Exercises – III - Solutions

Exercise 1: Hidden Markov Models

Imagine someone trying to deduce the weather from a piece of seaweed - folklore tells us that `soggy' seaweed means wet weather, while `dry' seaweed means sun. If it is in an intermediate state (`damp'), then we cannot be sure. However, the state of the weather is not restricted to the state of the seaweed, so we may say on the basis of an examination that the weather is probably raining or sunny. A second useful clue would be the state of the weather on the preceding day (or, at least, its probable state) - by combining knowledge about what happened yesterday with the observed seaweed state (soggish, damp, dryish, dry) we might come to a better forecast for today.

a) Design an HMM model (define its structure, number of states, hidden states, connectivity, variables to estimate, etc) that explains the observations, assuming that the hidden states (the true weather) are modelled by a simple first order Markov process. Draw the model's structure.

Solution:
Here is an example of model. Other solutions are also possible.



The variables to estimate are the observations, transitions and initial probabilities. Note that the labels for the hidden states in the figure above are just indicative of what we would hope to get after training. To check that the HMM with only 3 states resulted in a good segmentation across the 3 types of weather (sunny, cloudy and rainy), one should perform crossvalidation. During testing, one would first use the decoding step to determine the most likely sequence of hidden states to explain each of the testing observations. Then, one could see how many of these states correspond to one of the 3 true labels (assuming we know these). In case the match is poor, one may consider using a two-layer HMM (several layers HMM are called hierarchical HMM-s) with N intermediary states and 3 states in the last layer; the hope would be that the intermediary layer would encapsulate variations with the sunny, cloudy and rainy days and the last layer would. There, again, crossvalidation should be done to test that the last layer composed

Alternatively, one could use BIC to determine automatically the optimal number of states and hopefully discover that 3 states is enough. If one was to find more, one should do a careful readout of what each state has encapsulated by generating data from that state using the associated probabibility distribution of observations.

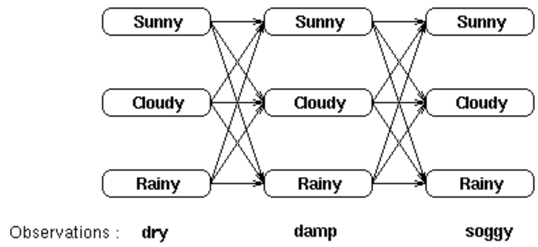
Note that one could also consider training separately the three states in the above figure with data from purely so-called sunny, cloudy and rainy days, and then train solely the transition probability across the states to encapsulate the temporal dependencies. This would however imply that one knows beforehand the labelling, which is rarely the case.

Remember that the strength of HMM is to discover the hidden structure behind a complex sequence of data and hence they should be used primarily for these types of applications. However, good practice is to try to understand what the hidden states encapsulate, whenever possible. The above discussion gives you some hint as to how to approach this problem.

b) Show the unfolding in time of the observation and state sequence used, e.g. in the Forward estimation procedure.

Solution:

Using the above three states model gives:



c) Explain how you can make an estimate of what the weather was for a week given each day's seaweed observation.

Answer: Again assuming that (a) yielded a 3-state model where each state corresponds roughly to what one would call a sunny, rainy and cloudy day, one would simply proceed to the Decoding Step, i.e. run the Viterbi Algorithm to estimate the most likely sequence of state, given the model and the observations. This will yield the sequence of sunny, cloudy, rainy days, etc.

d) Given a 1-week long sequence of seaweed observations, explain how you can use the model to determine whether it is winter or summer. Intuitively, if the seaweed has been dry for a while it may be summer; if it has been soggy for a while it might be winter.

Solution:
One could simply compute the number of times the "dry" and "soggy" states were inferred using the decoding step above and apply an argmax to decide.

e) Explain how you could adapt your model to learn transition across summer and winter based on a year-long set of seaweed observations

Solution

Intuitively, one would first think of using a two state Hidden Markov Model, assuming that each state would encode the distributions of weather observations for Summer and Winter, respectively. Following this rationale, one would propose to train the two-state model's parameters with the Baum-Welch procedure. Then, to determine whether one is in winter or summer (as per question d), one would proceed to the Decoding Step, i.e. run the Viterbi Algorithm to estimate the most likely sequence of states, given the model and the sequences of observation over a week. The last state of the sequence would then yield the answer.

The above reasoning is incorrect, because data are not labelled as winter and summer during training in the classical Baum-Welch learning procedure, and hence states are associated to group of observations (i.e. group of data), automatically to best describe the sequence of observation. Hence if there is more variability within the distribution of "winter" observations than across the "winter" and "summer" observations, the algorithm may end-up with a mixed (winter-summer) representation of the weather distribution for one or both states that would encapsulate other variability within the data than the intended one (e.g. the beginning of summer and winter were very rainy while the end of summer and winter were dry and hence early, temperature raises and decreases across both winter and summer, albeit by different amount; each state may encode separately for) not allow overlap in the prediction for summer and winter data for the two states.

To ensure that the two-state assumption is correct, one should proceed as recommended in (a) above, or consider adding more layers.

<u>Exercise 2 (Typical Exam Question!):</u>

Imagine that you have by now graduated from EPFL and work as a consultant for a car manufacturer. The car manufacturer sells 5 car models in 10 different countries. It has gathered data on its sales for the past 5 years and runs each year large pools to estimate its clients' tastes and level of satisfaction.

<u>I)</u> What would you recommend to your client to solve the following problems?

<u>a)</u> To predict the sales in each country for the next 5 years.

<u>b)</u> To determine which aspects of a car (speed and driving performance, aesthetic, costs, etc.) are most important to your client to help you design a new car model.

For each problem, suggest <u>one or more</u> Machine Learning algorithm(s) that could be used to solve the problem, and explain how it/they would solve it. That is, in each case, explain what the inputs and outputs of the algorithm would be (i.e. how the data are encoded), what cost function or bias you will use (only if applicable to your algorithm). Justify the use of this algorithm by explaining what type of computation the algorithm performs (classification, decorrelation, optimization, etc.) and why it is relevant for your problem. Draw a diagram that shows the flow of information with the variables. If the algorithm requires a preprocessing of the input, mention it explicitly.

<u>Solutions:</u>

<u>a)</u>

<u>!! This is only a partial answer. In an exam, you should be more detailed, give the list of variables and specify what you get input and output!</u>

This is a time-series. To predict the unfolding in time of the variables, one could use a Hidden Markov Model. In this case, the input would consist of …. Alternatively, one could use a non-linear regression algorithm (SVR or GMR) with time as one input dimension …[For a complete solution one should give more details on the actual implementation (number of states, what are the observables, what algorithm should be used to learn the data and retrieve the required predictions, c.f. slides and lecture notes of the course].

<u>b)</u>

E.g. run PCA or linear regression to determine which subset of variables matter most and whether there are some correlations across the data, e.g. people who buy red cars want these to also have powerful motors, whereas people purchasing white cars privilege safety.

One could also imagine performing classification on most popular, least popular type of cars based on current data to train the classifier. Then, one could do a grid search on all possible attributes and see which sets would be classified as popular car to generate a set of possible new model. The final choice would be based on cost. When using SVM, one could exploit distance to the margin to determine a measure of the chances that the new model will be popular (the farther away from the margin, the more likely). When using GMM, one could use the exact likelihood as a measure.