PR414 / PR813 Lecture 7 — Introduction to the Hidden Markov Model

This document is also available in PDF[1] format. Also look at this very handy HMM intro[2] on the WEB.

Purpose: To introduce the basic HMM concepts.

Material: Paper by Rabiner and Juang; book by Deller et al; paper by SE Levinson, "Structural methods in automatic speech recognition", Proceedings of the IEEE, vol 73 pp 1625 - 1650, Nov 1985.

General: Up until now the context of a series of feature vectors has been utilised in an overly simplistic way. It will be remembered from earlier lectures that we were able to vastly improve recognition accuracy by looking at a vector in context instead of in isolation. The assumption that we used, namely Independent with Identical Distribution (IID) is, however, often unrealistic. This is especially true in speech applications. (Were this not so, there would not have been much interesting information content in speech... Mmmm, now this can make one reconsider, maybe IID is appropriate after all! We'll have to do some experiments to settle this.)

With hidden Markov models (HMMs) this assumption is relaxed to include different distributions, as well as providing for some dependency between vectors. This technique can truly be said to form the backbone of current speech processing technology. Its implementation is fairly efficient and it yields fairly good results.

Topics:

- **The model** is a finite state machine which, when coupled with transition probabilities, results in a Markov Model. The symbols (features) that it emits is guided by a second set of distributions. This doubly stochastic process is the reason for the hidden part in the HMM. In the article are some examples which should make this clearer. Note that the given article is mostly concerned with the so-called discrete HMM, where the features are a set of discrete symbols. We will pay more attention to the (more accurate) case where they are continuous feature vectors.
- **Two assumptions** are being made:
  - The probability of being in a certain state at time t, is only dependent on the state active at time $t-1$.
  - The feature vectors are conditionally independent, with the condition being the particular state.

  Together these assumptions result in a system where the previous vector directly influences the current one. Indirectly this causes all vectors to influence each other.
- **Classification:** Two options are presented here, namely the forward-backward recursion, as well as the Viterbi recursion. The first is the theoretical precise solution and therefore also more accurate. It is somewhat slower than the Viterbi solution, and also suffers from numeric over/underflow problems. The Viterbi algorithm, which we will use, is fast and has no numerical problems. As an extra bonus it also yields the optimal state sequence associated with the features.
- **Training:** The Baum-Welch technique utilises the forward-backward recursions. The technique that we are going to use is, however, based on the Viterbi algorithm. It is not discussed in this article, but functions very much like the K-means clustering algorithm we saw in a previous lecture. It starts out with an initial model, segments the data according to it, and then updates the model according to the segmentation. This is repeated until convergence. See Deller p. 704 and p. 708 for details. A version of this for discrete HMMs is also given in the Levinson article.

[1] `http://www.dsp.sun.ac.za/pr813/lectures/7_hmm_a/7_hmm_a.pdf`
[2] `http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/main.html`

- **Initialisation:** It is quite beneficial to start the training process with reasonable initial conditions. Typically transition probabilities can be set to some plausible initial value. Density functions can be initialised in a number of ways:
  - If time-aligned transcriptions are available for the training data, this can be directly used to initialise the PDFs.
  - With ergodic HMMs it is useful to first cluster the data with a K-means algorithm, after which the initial PDFs are formed from these clusters.
  - PDFs in HMMs with a left-to-right structure can be initialised by dividing the data among the states on an equal length basis. This grouping is then used to set up the initial PDFs.

Task (hand in 3 lectures from now):

- (PR414) In the simvowel data, have a look at the wordx/y/z sets. These represent three different simulated words. Use the first 15 to 20 of each as training data and train three hmms. Then use the remaining wrd? data as the testing data and classify it. Compare the results against using GMMs instead of HMMs.
- (PR813, optional for PR414) Use the set of available diphthongs (phoneme codes `ey`, `aw`, `ay`, `oy` and `ow`) from the TIMIT data set to perform phoneme recognition (see `phoncode.doc` for pronunciation and other info). A Matlab function `timitphon.m`[3] is provided to extract sets of phonemes from the TIMIT data set. As an aside, this function is a good example of what a Matlab function should look like — pick up some tips from it! Train a 3-state left-to-right HMM with Gaussian output densities for each diphthong, using the same training (`sx`) data as before. Classify the remaining data (`si`,`sa`) using these models. Repeat the experiment using one of the prior static classification techniques, e.g. a Bayesian classifier with mixture Gaussian densities, and discuss.

  OR

  Figure out a way to use HMMs for image recognition. Describe each image either as a vertical sequence of 30 32-dimensional row feature vectors or a horizontal sequence of 32 30-dimensional column feature vectors. Then train a 5-state left-to-right HMM with Gaussian output densities for each class. Consider pose recognition instead of person recognition, in order to have more training data available (i.e. recognise the four classes `straight`, `up`, `left` and `right`). For pose recognition, a horizontal image sequence might work better. Diagonal covariance Gaussian output PDFs are probably a good idea. Repeat the experiment using one of the prior static classification techniques, e.g. a Bayesian classifier with mixture Gaussian densities, and discuss.
- *(Optional)* Evaluate the effectiveness of PCA/LDA in this situation by comparing HMM output PDFs with full covariance, diagonal covariance and diagonal covariance combined with KL decorrelation.

[3] `http://www.dsp.sun.ac.za/pr813/data/timitphon.m`