Hidden Markov Model in Music Information Retrieval
MUMT 611, March 2005
Assignment 5
Paul Kolesnik

Introduction to HMM

Hidden Markov Model (HMM) is a structure that is used to statistically characterize the behavior of sequences of event observations. By definition, HMM is "a double stochastic process with an underlying stochastic process which is not observable, but can only be observed through another set of stochastic process that produces the sequence of observed symbols" (Rabiner and Huang 1986).

The main idea behind HMM is that any observable sequence of events can be represented as a succession of states, with each state representing a grouped portion of the observation values and containing its features in a statistical form. The HMM keeps track of what state the sequence will start in, what state-to-state transitions are likely to take place, and what values are likely to occur in each state. The corresponding model parameters are an array of initial state probabilities, a matrix of state-to-state transitional probabilities, and a matrix of state output probabilities. The two basic HMM model types are an ergodic model, where any-to-any state transitions are allowed, and a left-to-right model, where state transitions can only take place to the state itself or to the subsequent state.

HMM delas with three basic problems—recognition, uncovering of states and training. The recognition problem can be formulated as: "given an observation sequence and a Hidden Markov Model, calculate the probability that the model would produce this observation sequence". The uncovering of states problem is: "given an observation sequence and a Hidden Markov Model, calculate the optimal sequence of states that would maximize the likelihood of the HMM producing the observation". The training problem states: "given an observation sequence (or a set of observation sequences and a Hidden Markov Model, adjust the model parameters, so that probability of the model is maximized". Through the algorithms used to solve those problems (Forward-Backward, Viterbi and Baum-Welch algorithms), an HMM can be trained with a number of observations, and then be used either for calculation of probability of an input sequence, or for identification of states in the input sequence interpreted by the HMM.

Overview of Works

This section provides an overview of the works presented at the International Symposium on Musical Information and Retrieval (ISMIR) that dealt with HMM-based systems.

A publication by Batlle and Cano (2000) describes a system that uses HMMs to classify audio segments. Using the system, audio files are automatically segmentated into abstract acoustic events, with similar events given the same label to be used for training of the HMMs. THe system is applied to classification of a database of audio sounds, and allows fast indexing and retrieval of audio fragments from the database. During the initial stage of the process, mel-cepstrum analysis is used to obtain feature vectors from the audio information, which are then supplied to an HMM-based classification engine. Since traditional HMMs are not suited for blind learning (which is the goal of this system, as there is no prior knowledge of the feature vector data), competitive HMMs (CoHMMs) are used instead. CoHMMs differ from HMMs only in training stage, whereas the recognition procedure is exactly the same for both,

A work by Durey and Clements (2001) deals with a melody-based database song retrieval system. The system uses melody-spotting procedure adopted from word spotting techniques in automatic speech recognition. Humming, whistling or keyboard are allowed as input. According to the publication, the main goal of the work was to develop a practical system for non-symbolic (audio) music representation. The word/melody-spotting techniques involve searching for a data segment in a data stream using HMM models. Left-to-right, 5-state HMMs are used to represent each available note and a rest. As part of the preprocessing process, frequency and time-domain features are used for extraction of feature vectors. The system records the extracted feature vectors for all of the musical pieces and stores them in a database. Once an input query is received, it constructs an HMM model from it, and runs all of the feature vectors from the songs in the database through the model using the Viterbi state uncovering process. As a result, a ranked list of melody occurrences in database songs is created, which allows identifying the occurrence of the melodies within the songs.

A publication by Jin and Jagadish (2002) describes a new technique suggested for HMM-based music retrieval systems. The paper describes traditional MIR HMM techniques as effective but not efficient, and suggests a more efficient mechanism to index the HMMs in the database. In it, each state is represented by an interval / inter onset interval ratio, and each transition is transformed into a 4-dimensional box. All boxes are inserted into an R-tree, an indexing structure for multidimensional data, and HMMs are ranked by the number of boxes present in the search tree. The most likely candidates from the R-tree are selected for evaluation of the HMMs, which uses the traditional forward algorithm.

A publication by Orio and Sette (2003) describes an HMM-based approach to transcription of musical queries. HMMs are used to model features related to singing voice. A sung query is considered as an observation of an unknown process—the melody the user has in mind. A two-level HMM is suggested for pitch tracking: an event-level (using pitches as labels), and audio-level (attack-sustain-rest events).

A paper by Sheh and Ellis (2003) deals with the system that uses HMMs for chord recognition. An EM (Expectation-Maximization) algorithm is used to train the HMMs, and PCP (Pitch Class Profile) vectors are used as features for the training process. Each chord type is represented by an independent HMM model. According to the publication, the system is able to successfully recognize chords in unstructured, polyphonic, multi-timbre audio.

Shifrin and Burmingham (2003) present a system that investigates performance of an HMM-based query-by-humming system on a large musical database. A VocalSearch system, that has been desgined as part of MusArt project, is used on a database of 50000 themes that have been extracted from 22000 songs. The system uses a <delta-pitch / Inter-onset Interval ratio> pair as parameters for the feature vectors supplied to the HMMs. The work compares perfect queries with imperfect queries that have simulated insertions and deletions added to the sequences. Some of the trends discovered as a result of the work are: longer queries have a positive effect on evaluation performance, but all experiments show an early saturation point where performance does not improve with query length. The system performed well with imperfect queries on a large database.

Bibliography

Batlle, E., and P. Cano. 2000. Automatic segmentation for music classification using Competitive Hidden Markov Models. In *Proceedings of International Symposium on Music Information Retrieval*.

Durey, A., and M. Clements. 2001. Melody spotting using Hidden Markov Models. In *Proceedings of International Symposium on Music Information Retrieval,* 109–17.

Jin, H., and H. Jagadish. 2002. Indexing Hidden Markov Models for music retrieval. In *Proceedings of International Symposium on Music Information Retrieval,* 20–4.

Orio, N., and M. Sette. 2003. A HMM-based pitch tracker for audio queries. In *Proceedings of International Symposium on Music Information Retrieval,* 249–50.

Rabiner, L., and B. Huang. 1986. An introduction to Hidden Markov Models. IEEE Acoustics, Speech and Signal Processing Magazine 3(1), 4–16.

Sheh, A., and D. Ellis. 2003. Chord segmentation and recognition using EM-trained Hidden Markov Models. In *Proceedings of International Symposium on Music Information Retrieval,* 183–9.

Shifrin, J., and W. Burmingham. 2003. Effectiveness of HMM-based retrieval on large databases. In *Proceedings of International Symposium on Music Information Retrieval,* 33–9.