Music Informatics

Alan Smaill

Feb 2, 2017

Organisation

The courseworks are based on the same topic, so feedback from the first will help you in the second.

- First coursework (formative: feedback, no mark recorded)
    - Issued Monday Feb 6th
    - Due Monday Feb 28th
- Second coursework (summative: mark recorded, feedback given)
    - Issued Thursday Feb 27th
    - Due Thursday March 24th

Today

- Score Following
- real time vs offline
- input: midi vs audio
- use of hidden Markov models

Score Following

This is a problem related to beat tracking, but different because we assume that a musician is performing from a score, and we want to be able to track the musician's progress through the score.

Typically the aim is to follow a soloist so as to coordinate something produced artificially with the soloist's playing.

The soloist should be able to play the music freely, and not be forced to follow any rigid tempo, and also add small variations to the score without the system losing track.

Some versions of the problem

As in beat tracking, there are different versions of the problem:

- Input: midi-like or audio
  as usual midi helps where it is easy to get this sort of input
- Processing: real time, or off-line;
  again real-time is what is wanted for performance, but harder to achieve, and not necessary for some tasks.

Score following is used eg for

- coordination of accompaniment to a soloist automatically, involving prediction of when the next accompaniment notes should appear;
- in compositions with a mixture of human performers and virtual musicians following a score.

Score Following

The best outcome is a system that will listen, perform and learn ...

> *to recognise the computer's potential not as a simple amplifier of low-level switching or acoustic information (keyboards and live audio distortion), but as an intelligent and musically informed collaborator in live performance as human enquiry.*
>
> *Vercoe,*
> *The Synthetic Performer in the Context of Live Performance, Proc ICMC 1984*

HMM Approach

A widely used approach is to use some version of Hidden Markov models.

- the productions are musical events (notes, usually);
- the hidden states correspond to places in the score;
- the score provides transition probabilities that allow for some flexibility (eg missing, extra or wrong notes).

An HMM system can be used to estimate the most likely position in the score, given a sequence of productions.
It can also learn improved transition probabilities to adapt to a particular performer.

HMM basics

There are many places on-line to look to find out about HMMs: the classic reference is:

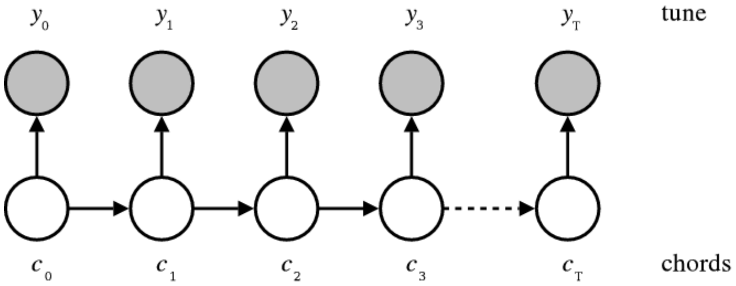
`www.cs.ubc.ca/~murphyk/Bayes/rabiner.pdf`

Markov Models (MMs) provide a way to represent sequential data (items ordered as a sequence),
so works where the data has a natural linear order, e.g. notes from monophonic instrument.

The "hidden" variety (HMMs) suppose that the observed productions come with certain probabilities from hidden states, and that there are transitions between the hidden states, also with associated probabilities.

Basic HMM

In this example, “chord” is hidden, “tune” (melody note) is observed (used for harmonisation):



Assigning Probabilities

An HMM also associates probabilities with transitions and emissions:

- In hidden state, what is the probability of a given observation?
- In hidden state with different possible successor states, what are the probabilities of moving to another state?

These can be assigned by hand, or the network can be trained and made to model phenomena that typically have this linear character.
The most important notion is that of conditional probability:
what is the probability of some event(s) E given that some other event(s) F has been observed:

this is written as $P(E|F)$.

Running an HMM

A run of a given HMM with given transition and emission probabilities works like this:

- Start in initial state at time $t = 1$ (might be randomised also)
- Generate first observation according to emission probabilities
- Move according to probabilities to some next state if possible, and increment time (otherwise terminate)
- Iterate appropriately.

HMM tasks and algorithms

There are 3 standard tasks with associated computations. Suppose we have a given HMM H, and a sequence of observations $O = O_1, \ldots, O_n$.

- What is the probability $P(O|H)$ that it was generated by the HMM?
- What is the sequence of hidden states that gives the best account of the observations?
- How can we adjust the probability parameters of H to maximise $P(O|H)$?

The last is treated by a machine learning algorithm; the aim is to create "best models for real phenomena" (Rabiner).

Deviation from score

See papers on score following by Orio and associates at IRCAM, and bunch of resources, including excepts of performances using the technology:

http://imtr.ircam.fr/imtr/Score_Following

The most recent score following system from IRCAM takes a different approach however, we will look at examples later. The approach described here can be seen at work in this clip:

https://youtu.be/q55Okme1vTc

Ghost states

A way to allow for different paths related to the score is to include some "ghost" hidden states, to allow for

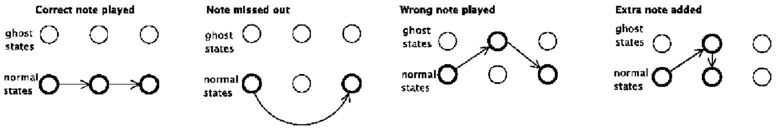
- wrong notes;
- extra notes; and for
- skipped notes.

Give these a states lower probability than that of the expected outcome (see paper by Orio and Déchelle).

Achieve this by having a ghost state for each expected state, with associated productions:

- wrong note: go to ghost, then to expected next plus one
- extra note: go to ghost, then to expected next
- missing note: go to ghost, then to next plus two.

HMM design



The Orio and Déchelle paper also uses HMM at the level of the audio signal. This is an instance found fairly often of using HMM at two different levels.
Individual notes pass through a typical sequence of attack – sustain – possible silence at the end. This can also be modelled by an HMM, allowing the hidden “sustain” state to have a good chance of a transition back to itself.

Music plus one

Christopher Raphael has worked for some time on an impressive approach to score following in the context of accompanying a solo musician playing classical music.
The emphasis is on dealing with the normal shaping of tempo, and allows for addition of ornaments such as trills without problem.
See the web site at

`http://xavier.informatics.indiana.edu/~craphael/`

Music Plus One

Raphael says:

> *Specifically, my goals are that the program must respond in real time to the soloist's tempo changes and expressive gestures; the program must learn from past performances so that it assimilates the soloist's interpretation in future renditions; and it must bring a sense of musicality to the performance in addition to what is learned from the soloist. In this way MPO *adds* to the soloist's experience by providing a responsive and nuanced accompaniment rather than *subtracting* from it by imposing a rigid framework that stifles musical expression.*

Architecture

. . . is in terms of interleaved listen/play processes.

- The input is digitised audio signal, processed in real time.
- The performance is modelled as an HMM, where hidden states relate to the score, including timing information; and observables correspond to (analysed features of) the audio
- This is interleaved with the generation of accompaniment, that uses the best guess at position in the score to predict when to output some part of the accompaniment.

Learning to play

The accompaniment process has to merge:

- the output of the listen process
- an analytic understanding of the musical score;
  the system deals with standard musical readings of rhythm, phrasing, and tempo correspondences
- collection of previous performances by given soloist
- performances of the accompaniment by a live player

Playing via Bayesian belief network

This is another related technology based on probability.
Raphael uses this in a large system to allow learning by the playing system, as described above. It incorporates variables for local tempo and so on, that inform the estimation of the soloist's interpretation of the piece.
The arcs in such a network make assumptions about statistical independence between variables. It is computationally necessary for efficiency reasons to make such assumptions, when as here there are hundreds of variables at issue; Raphael argues that these independence claims are musically plausible.

Rehearsal

During a rehearsal phase, the network can be trained to model the interpretations demonstrated by the solo and accompaniment examples, thereby allowing the system to automatically adapt to each new musical situation. Once this is done, the network provides the basis for a real-time performance engine that guides the accompaniment through a course of actions, each informed by all currently-available knowledge.

C Raphael

Listen to some of the examples, and also the short movie.

Summary

- score following
- versions depending on input and real-time
- using HMM to model score location
- responsive and learning automatic accompaniment