MUMT 611

Final project:

Classification of melody by composer using hidden Markov models

Greg Eustace

Introduction:

The goal of this project is to use Hidden Markov Models (HMM) for the automatic classification of symbolic melodic data by composer. Several research questions are posed as part of this work. Are there significant statistical differences between melodies written by different composers? How do different representations of melody affect the performance of the classifier? How do different types of HMMs affect the performance?

Representations of melody:

Melody may be defined as a monophonic succession of tones characterized by information pertaining to pitch and rhythm. In polyphonic music the melody is defined as the dominant tune, which may or may not be sounded by a single voice or instrument. The melody is typically represented by the voice or voices occupying the highest range.

Alternative representations of melody may effect classification. It has been shown that the features most perceptually relevant to melody identification are pitch and rhythm combined, followed by pitch alone and rhythm alone (Uitdenbgerd, 1999). Although certain aspects (e.g. pitch) may be more perceptually salient alone, reducing the representation of melody to a single aspect may present the researcher with false positives.

Absolute pitch is readily represented by a sequence of integers, such as MIDI note numbers, while pitch class can be represented by normalizing pitch data to the octave.

Intervals express the difference in pitch between two notes. The main advantage over interval vs. pitch representations is that intervals are transposition invariant. So called modulo interval techniques reduce intervals greater than 12 semitones by 12 (Uitdenbgerd, 1999).

Probably the most commonly implemented algorithms represent melody by melodic contour, in which an instance of a note is characterized by its relationship to the proceeding note. Typically three possible representations are allowed including: up, down and same. However, other representations are possible which group ascending and descending intervals into two or more categories.

Rhythmic information seems to be less often considered, even though it contributes to the perception of melody. Relevant aspects include note durations, the position of rests and stress.

It is perceptually consistent to weight certain notes in a melody according to their importance. Note duration is probably the most common measure of importance, where shorter notes receive a lower weighting.

Hidden Markov Models:

Hidden Markov models are used for the purpose of classification. This implementation uses the HMM Toolbox by Kevin Murphy, which is available for free at: http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html.

Rabiner (1989) defines a Hidden Markov model as a doubly imbedded stochastic process with an underlying stochastic process that is not observable (i.e. is hidden) but can only be observed through another set of stochastic processes that produce the sequence of observations. HMMs are typically defined by five parameters including the number of distinct observation symbols, the number of states in the model, the state transition probability distribution, the observation symbol probability distribution and the initial state distribution (Rabiner 1989).

Two basic kinds of HMMs are fully connected (ergodic) models in which every state is connected to every other state and left-right (Bakis) models in which states are connected to themselves and to the adjacent state, proceeding from left to right (Rabiner 1989). Left-right models are typically used for modeling time varying signals (Rabiner 1989). Chai and Vercoe (2001) use three variations of the left-right model. The first type is a strict left-right model. For type two models each state can transfer to its self and any state to the right of it. Type three models are identical to the second type except that the last state can transfer to the first state.

The Baum-Welch learning algorithm is used to find the hidden parameters of the HMM. This process uses maximum likelihood parameter estimation. In general, the likelihood is maximized when a given test sequence corresponds to a specific model. It is also common to attempt to maximize the logarithm of the likelihood. The process of expectation maximization (EM) accepts a random guess as to what the hidden parameters should be and improves upon this guess iteratively.

Preprocessing:

The raw data set consisted of thirty type I MIDI files. These included all fifteen of Bach's two-part inventions as well as fifteen of Chopin's Nocturnes. Of the Nocturnes numbers 1 to 10, 13 to 15, 18 and 21 were chosen as they contained the least amount of polyphony in the first two bars. Each file consisted of two tracks representing left-hand and right-hand parts.

Several preprocessing stages were automated using a MIDI sequencing program. First the right hand parts were extracted from each file. The nocturnes contained some polyphony, in which case all notes not comprising the soprano line were deleted as it was required that the data be monophonic. 16^{th} notes appeared frequently in the data while notes of smaller durations were less common. Thus, all notes smaller than a 16^{th} note were deleted. This constraint would seem to be in keeping with perception, because notes with smaller durations tend not to contribute as much to the perception of melody as larger ones. Notes were also quantized, in terms of both note length and position, to the nearest 16^{th} note. This constraint removes rhythmic articulations, transposition mistakes and tuplets. Tuplets are mostly encountered in the Nocturnes. Tempo articulations were removed and the data set was transposed to 60 BPM. In addition, all files were transposed to the key of C major.

At this point, the MIDI note numbers and note durations were extracted using MAX/MSP and imported to MATLAB for additional pre-processing. All information pertaining to rests was decimated. Notes which precede a rest are assumed to be extended in duration by the length of the rest.

Preprocessing in MATLAB consists of several steps. Pitch data are converted to interval and contour representations by the users specification. The representation for contour included descending leaps, steps and no change, followed by ascending leaps and steps. These states are represented by the numbers 1 to 5 respectively. For each melodic representation the data were normalized to the octave. Thus, pitch, interval and contour representations consist of 12, 24 and 5 observations respectively. In all cases the first observation is one, because this is the first number in the discrete alphabet. At this point rhythmic information can be added. Chai and Vercoe (2001) suggest repeating notes of larger duration in accordance with their relationship to the smallest note present. In this case 16th notes are not repeated, 8th notes are repeated twice, quarter notes are repeated four times, and so forth.

At this point the data set can be formed from the first 16 or 32 notes from each of the thirty data vectors (15 or 31 for interval or contour representations). Thus, for rhythmic representations the data pertain to the first 1 or 2 bars, where as for non-rhythmic representations the data pertain to the first 16 or 32 notes. In accordance, the number of states of the HMM is specified to be either 15, 16, 31 or 32. Both Bach and Chopin data sets can then be separated into a training set and a testing set. Training and testing data must have the same number of columns, so they are spilt along the row dimension according to the closeness to a user specified percentage.

Implementation and Results:

The classification process is summarized by four steps that include: training a different model on data from each composer (handled by the script dhmm_em.m); computing the log-likelihood that each test sequence belongs to each model (handled by the script dhmm_logprob.m); classifying the training data according to the model which gives the highest value for the log-likelihood; repeating the process for all representations of melody and HMM types. In order to perform the classification you must install the HMMtoolbox and run the MATLAB script classify.m.

One additional preprocessing step was necessary. It is a requirement of the training procedure that all observations in the testing set be observed in the training set. This is an unfortunate constraint as it leaves two options: either cherry picking training and testing sets which conform to this constraint or adding erroneous values to the training set to compensate. The first option may be viable if a certain percentage (or higher) of the data set is used for testing. For the data set in question, it was decided to add an additional column of data to the training set consisting of the missing observations. This was not a necessary step for contour data. Thus, for pitch and interval data 3 % of the data are false. In case the number of missing observations was not equal to the column length, the rest of the column was filled with data from the first column. It should also be noted that if an extract column was added the number of states were increased by one to reflect this.

The log likelihoods for data trained and tested on both Chopin and Bach are given in the appendix. The resutlts were calculated using five iterations of EM. Although the actual values for the log likelihoods varied according to the number of iterations, running greater iterations did not appear to affect classification.

Contour without rhythm was the only representation to consistently classify both test sets for both training sets, but only when the data set was formed from the first two bars. This seems in accordance with the importance of contour as a perceptually salient representation of melody. However, it may also suggest that the HMM performs better for fewer observations. It is also clear that interval and pitch data were slightly contaminated, and this has had some affect.

The results concur with the findings of Chai and Vercoe (2001) that this type of rhythmic representation does not facilitate better classification results. However, it is surprising that in no case did the specification of model type affect the classification process significantly. I am quite suspicious of this finding. Furthermore, it is clear that a much larger data set should be used in order to verify the validity of these results.

Appendix:

Training set: Bach (66.6%); Testing set: Bach (33.3%)								
Rep.	First bar				First two bars			
	Full	L/R 1	L/R 2	L/R 3	Full	L/R 1	L/R 2	L/R 3
Pitch	-163.12	-159.40	-157.96	-158.26	-369.18	-409.82	-348.12	-347.36
P & R	-157.66	-180.42	-150.94	-151.36	-325.70	-406.26	-323.90	-321.38
Interval	-225.99	-234.12	-215.70	-212.95	-403.53	-473.55	-403.09	-400.02
I & R	-247.43	-252.85	-232.05	-233.40	-421.20	-475.50	-441.90	-441.44
Contour	-125.65	-132.79	-124.81	-121.35	-224.70	-263.84	-222.03	-218.06
C & R	-130.07	-151.45	-128.60	-125.25	-234.76	-248.29	-230.24	-232.01

Training set: Bach (66%); Testing set: Chopin (33%)								
Represent.	First bar				First two bars			
	Full	L/R 1	L/R 2	L/R 3	Full	L/R 1	L/R 2	L/R 3
Pitch	-222.61	-228.66	-218.38	-216.58	-450.34	-469.36	-440.76	-436.44
P & Rhy.	-205.74	-223.23	-211.42	-215.97	-400.01	-433.02	-416.50	-414.83
Interval	-204.11	-217.76	-195.15	-194.03	-405.72	-452.71	-397.06	-393.26
I & Rhy	-194.99	-186.89	-187.48	-190.32	-383.57	-424.91	-394.43	-399.47
Contour	-121.80	-126.72	-123.52	-119.34	-268.11	-287.29	-237.94	-236.92
C & Rhy	-85.72	-111.92	-90.71	-90.62	-200.54	-216.97	-185.73	-194.32

Training set: Chopin (66%); Testing set: Chopin (33%)								
Represent.	First bar				First two bars			
	Full	L/R 1	L/R 2	L/R 3	Full	L/R 1	L/R 2	L/R 3
Pitch	-186.42	-190.50	-176.66	-169.44	-375.84	-395.47	-365.86	-366.04
P & Rhy.	-194.20	-225.78	-164.98	-167.52	-321.98	-370.18	-325.36	-334.16
Interval	-214.09	-233.76	-211.74	-209.69	N/A	N/A	N/A	N/A
I & Rhy	-268.31	-253.82	-238.32	-223.31	-472.79	-454.07	-419.96	-416.89
Contour	-128.67	-137.89	-120.06	-120.25	-252.91	-307.10	-231.63	-231.15
C & Rhy	-151.36	-144.09	-118.81	-119.12	-236.31	-270.29	-207.19	-206.73

Training set: Chopin (66%); Testing set: Bach (33%)								
Represent.	First bar				First two bars			
	Full	L/R 1	L/R 2	L/R 3	Full	L/R 1	L/R 2	L/R 3
Pitch	-234.92	-257.07	-229.28	-219.53	-449.35	-467.40	-435.60	-434.40
P & Rhy.	-213.80	-284.49	-201.68	-215.68	-369.02	-386.68	-349.15	-374.18
Interval	-178.15	-194.90	-184.34	-182.43	N/A	N/A	N/A	N/A
I & Rhy	-181.12	-168.09	-154.96	-160.31	-345.23	-344.14	-298.97	-301.89
Contour	-128.80	-131.74	-116.58	-115.53	-233.52	-302.06	-227.27	-223.18
C & Rhy	-112.75	-82.23	-71.13	-70.70	-220.71	-254.71	-159.49	-166.83

References:

Chai, W., and B. Vercoe. *Folk music classification using hidden Markov models. 2001*. Proceedings of the international conference on artificial intelligence.

Rabiner, L. 1989. *A tutorial on hidden Markov models and selected applications in speech recognition*. Proceedings of the IEEE 77: 257–86.

Uitdenbgerd, A., and J. Zobel. 1999. Melodic matching techniques for large music databases. *Proceedings of the Seventh ACM International Conference on Multimedia*: 57–66.