Classifying Polyphonic Melodies By Chord Estimation Based on Hidden Markov Model

Yukiteru YOSHIHARA† Takao MIURA† Isamu SHIOYA‡

† Dept.of Elect.& Elect. Engr., HOSEI University
3-7-2 KajinoCho, Koganei, Tokyo, 184–8584 Japan
‡ Dept.of Informatics, SANNO University
Kamikasuya 1573, Isehara, Kanagawa, Japan

Abstract. In this investigation we propose a novel approach for classifying polyphonic melodies. Our main idea comes from for automatic classification of polyphonic melodies by *Hidden Markov model* where the states correspond to well-tempered chords over the music and the observation sequences to some feature values called *pitch spectrum.* The similarity among harmonies can be considered by means of the features and well-tempered chords. We show the effectiveness and the usefulness of the approach by some experimental results.

Keywords: Melody Classification, Melody Features, Markov process, Hidden Markov Modeling

1 Motivation

We propose a novel approach for classifying polyphonic melodies. We define similarity among polyphonies in terms of *features*, and generate Hidden Markov Model (HMM) analyzing training music data and classifying any music with *Viterbi* algorithm. During the process, we estimate chord progression since state transition corresponds to chord progression in principle. Readers are assumed to be familiar with basic notions of music[5] and basic IR techniques[4].

Section 2 and 3 contain several definitions of features for melody description that have been proposed so far. In section 4 we review Hidden Markov models and discuss how to apply HMM to our issue of music classification. We show some experiments and some relevant works in section 6. Finally we conclude our investigation in section 7.

2 Melody and Polyphony

Music consists of many *tones*. Each tone consists of *pitch*, *duration* and *strength.* Pitch means modulation or height of tone that is defined by *frequency.* For example, 440Hz sounds like A (*la*). Two times height is called an *octave* thus 880Hz sounds like a, an octave higher pitch[1]. An *interval* is a distance between two

[1] We denote an octave pitch by small letter.

pitch. We equally divide one octave into 12 *semitones*. Given an octave between `A` and `a`, we obtain 12 semitones denoted by `A,A#,B,C,C#,D,D#,E,F,F#,G,G#`. The *well-tempered* sequence consists of 2,1,2,2,1,2,2 semitones or 2,2,1,2,2,2,1 semitones between adjacent pitch. The former sequence is called *minor* and the latter *major*. Also, in the sequence, `A`, `C` are called *keys*. *Transposition* means relative movement of keys while keeping the number of semitones between adjacent pitch of minor/major intervals.

Each tone has its *duration* which describes length aspect of the pitch. *Melody* is a sequence of tones, or pitch/duration while *chord* or *harmony* is description of tones in parallel which are assumed to be played simultaneously. *Theme* is an intended melody which composers think most important. Music except theme is called *accompaniment*. *Polyphony* is music with accompaniment while *monophony* is the one without. That is, in monophony, there can be at most one tone in music. *Five horizontal lines* contain expression which is described by a list of notes with pitch and duration in both sequence and parallel. The expression on the lines is called *score*. Here melody is described by a set of notes arranged into a sequence. *Beat* is a summarized duration in the notes, and a *bar* is a partial description in a score which contains equal size of beat. *Time of signature* means the number of the beat and its characteristic. Especially *rhythm* is a rule how beat and the characteristic is constructed.

3 Features for Melody Classification

To specify and classify melodies, we should examine what kinds of semantics they carry and we should describe them appropriately. Since we need score-based features for classification purpose, we should examine notes over score or in bars. We discuss several kinds of features, and we put these characteristic values into vector spaces[4]. Similarity between two melodies is defined by their *cosine value*. The similarity can be modeled by ranking these values[12].

First of all, let us describe several features for monophonic melody description[8]. *Melody Contour* is one of the major techniques proposed so far[1, 7, 11]. *Pitch Contour* is the one where we put stress on incremental transition of pitch information in monophonic melody. Unfortunately there exist many problems in pitch contour[11].

Given a melody on score, we introduce *Pitch Spectrum* per bar in the melody for similarity measure[8]. *Pitch Spectrum* is a histogram where each column represents total duration of a note within a bar. The spectrum constitutes a vector of $12 \times n$ dimensions for n octaves range. We calculate pitch spectrum to every bar and construct characteristic vectors prepared for querying music.

By pitch spectrum we can fix several problems against incomplete melody. In fact, the approach improves problems in swinging and grace. Note that score approach improves issues in rhythm, keys, timbre, expression, speed, rendition and strength aspects of music. Some of the deficiencies are how to solve transposition (relative keys) issues and how to distinguish majors from minors[8]. Especially the latter issue is hard to examine because we should recognize the contents.

On the features for polyphonic melodies, we assume that polyphonic scores are given in advance to obtain the feature values. Also, we extract all the tones in each bar from polyphonic music and put them into a spectrum in a form of vector. Since the pitch spectrum (for monophonic melody) has been generated as a histogram in which each column represents total duration of a note within the bar, it is straightforward to generate a (polyphonic) pitch spectrum from counting notes appeared in all the parts in the bar. Clearly the new spectrum reflects not only all the tones in the bar but also all the noises for classification like grace/trill notes. We take modulo 12 to all the notes (i.e., we ignore octave). Then we define *(polyphonic) pitch spectrum* as the pitch spectrum as mentioned that consists of only n biggest durations considered as a chord. If there exist more than n candidates, we select the n tones of the highest n pitch. Note that we select n tones as a chord but ignore their explicit duration. And we define the *feature description* of the length m as a sequence $w_1, ..., w_m$ where each feature w_j is extracted from i-th bar of music of interests.

EXAMPLE 1 Let us describe our running example "A Song of Frogs" in figure 1. Here are all the bars where each collection contains notes with the total duration counted the length of a quarter note as 1.

```
{C:1, D:1, E:1, F:1}, {C:2, D:2, E:2, F:1}, {C:1, D:1, E:2, F:1, G:1, A:1 }, {E:2, F:2,
G:2, A:1}, {C:2, E:1, F:1, G:1}, {C:4}, {C:3, D:1, E:1, F:1}, {C:2, D:2, E:2, F:1}
```

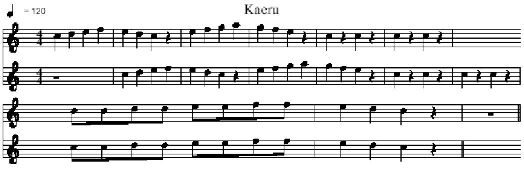

Fig. 1. Score of *A Song of Frogs.*

The sequence of the pitch spectrums constitute the new features for all the bars by top 3 tones. In this case we get the feature description (DEF, CDE, EGA, EFG) for the first 4 bars.

```
Bar1 : {C:1, D:1, E:1, F:1} = {DEF}
Bar2 : {C:2, D:2, E:2, F:1} = {CDE}
Bar3 : {C:1, D:1, E:2, F:1, G:1, A:1} = {EGA}
Bar4 : {E:2, F:2, G:2, A:1} = {EFG}
```

4 Hidden Markov Model

A *Hidden Markov Model* (HMM) is an automaton with output where both the state transition and the output are defined in a probabilistic manner. The state transition arises according to a simple Markov model but it is assumed that we don't know on which state we are standing now[2], and that we can observe an output symbol at each state. We could estimate the transition sequences through observing output sequence.

Formally a HMM model consists of (Q, Σ, A, B, π) defined below[6]:

(1) $Q = \{q_1, \cdots, q_N\}$ is a finite set of states
(2) $\Sigma = \{o_1, \cdots, o_M\}$ is a finite set of output symbols
(3) $A = \{a_{ij}, i, j = 1, ..., N\}$ is a probability matrix of state transition where each a_{ij} is a probability of the transition at q_i to q_j. Note $a_{i1} + ... + a_{iN} = 1.0$.
(4) $B = \{b_i(o_t), i = 1, ..., N, t = 1, ..., M\}$ is a probability of outputs where $b_i(o_t)$ is a probability of an output o_t at a state q_i
(5) $\pi = \{\pi_i\}$ is an initial probability where π_i is a probability of the initial state q_i

In this work, each state corresponds to a well-tempered chord such as $< C >$ = CEG, and the set of states depends on a polyphonic melody. Output symbols (pitch spectrum such as CDE) should be observable and identifiable in our case. Note spectrums do not always go well with well-tempered chords theoretically.

A HMM is suitable for estimation of *hidden* sequences of states by looking at observable symbols. Given a set of several parameters, we can obtain the state sequence which is the most likely to generate the output symbols. The process is called a *decoding problem* of HMM.

Here we define the most likely sequence of states as the one by which we obtain the highest probability of the output generation during the state transition. The procedure is called *Maximum Likelihood Estimation* (MLE). In the procedure, once we have both sequences of the states and the output symbols, we can determine the probabilities (or *likelihood*) of the state transition and of the output generation along with the state transition. Putting it more specifically, when we have the state transition $q_1 q_2 \cdots q_T$ and the output sequence $o_1 o_2 \cdots o_T$, we must have the likelihood as $\pi_{q_1} b_{q_1}(o_1) \times a_{q_1 q_2} b_{q_2}(o_2) \times \cdots \times a_{q_{T-1} q_T} b_{q_T}(o_T)$. A naive calculation process is called a *Forward* algorithm. *Viterbi* algorithm is another solution for the decoding problem. Given the output sequence $o_1 \cdots o_T$, the algorithm is useful for obtaining the most likely sequence of the states by taking the highest likelihood $\delta_t(j)$ of an output o_t at a state q_i to go one step further to q_j. That is, the algorithm goes recursively as $\delta_{t+1}(j) = \max_i(\delta_t(i) a_{ij}) b_j(o_{t+1})$. During the recursive calculation, we put the state q_j at each time t, and eventually we have the most likelihood sequence $q_1 \cdots q_T$.

In a HMM, there is an importnat issue, how to obtain initial HMM parameters A, B and π. One of the typical approach is *supervised learning.* In this approach, we assume *training data* in advance to calculate the model, but the data should be correctly classified by hands since we should extract typical patterns them by examining them. Another approach comes, called *unsupervised learning.* Assume we can't get training data but a mountain of unclassified data

[2] This is why we say *hidden.*

except a few. Once we obtain strong similarity between the classified data and unclassified data (such as high correlation), we could extend the training data in a framework of Expectation Maximization (EM) approach[6]. Here we take supervised learning by analysing scores.

5 HMM for Melody Classification

In this investigation, we assume a collection of music classes and we classify an unknown music d into one of the classes where $d = \{w_1, ..., w_m\}$ and w_j are a feature. In our approach, we consider well-tempered chords[5] as states, and given vectors of pitch spectrums considered as observable sequences, we estimate how the state transition arises and what class of music is most likely.

According to the general procedure of HMM, let us apply the HMM to our problems. We already know a set of well-tempered chords by theory of music[5] and a set of possible features (pitch spectrums). Given a collection $\mathcal{C}$ of polyphonic melodies with classes as training data and an unknown music d to be examined, we estimate a class label c_k of d.

(a) By examining $\mathcal{C}$, we generate a probability matrix A of state transition and probability B of symbol output at each state in a HMM model (Q, Σ, A, B, π) where Q means a set of well-tempered chords, Σ a set of possible features.
(b) Then we estimate a membership probability $P(c_i|d)$ of a class c_i by using the HMM model, $i = 1, ...$ Then, by Maximum Likelihood Estimation (MLE), we have $c_k = ArgMax_{c \in C} P(c|d)$.

Since we classify unknown music into one of the classes given in advance, we compare pitch spectrum with each other, but very often we have many chances to see no common chord among features such as { CDE, DEF } and { CEG, DEG }. Note we have constructed sequence of pitch spectrum in each bar[3]. During comparison of feature descriptions of two music d_1, d_2, it is likely to have some feature (a chord) w in d_1 but not in d_2 at all. In this case, the probability must be 0.0 in d_2 and the membership probability should be zero. Then the two music can't belong to a same class even if the most parts look much alike. Such situation may arise in the case of noises or trills.

To solve this problem[4], we introduce a notion of similarity between each pair of chords and we adjust the probabilities with them. We introduce a notion of *similarity* between two pitch spectrum and we adjust probability of output observation at each state. Assume there are $w_1, ..., w_k$ outputs at a state s, and we see an observation w. We define the probability $P(w_i|s)$ where we have an observation w_i at the state s. Then the observation probability of w at s, denoted by $P'(w|s)$, is adjusted as $P'(w|s) = \sum_i^k P(w_i|s) \times sim(w_i, w)$ where

[3] Remember pitch spectrum is a histogram in which each column represents total duration of a note within a bar and we select top n tones for the spectrum.

[4] Some sort of revisions are usually introduced which causes erroneous classification.

sim means *cosine* similarity. Given observations $w_1 ... w_k$ at states $s_1, ..., s_k$, we see the adjusted probability of the observation sequence $P'(w_1 ... w_k | s_1 ... s_k) = \prod_{i=1}^{k} P'(w_i | s_i)$ as usual.

EXAMPLE 2 First of all, we show the similarity V of two spectrums $\{CDE\}$ and $\{CDF\}$. Since we have two common tones C, D, $V = \frac{2}{\sqrt{3 \times 3}} = 0.66$.

Then let us illustrate our approach as running examples with chords of top 3 tones. We have *A Song of Frog* (d_1) with a label `A` in figure 1, and *Ah, Vous dirai-Je, Maman* in C Major, KV.265, by Mozart (d_2) with a label `B` in figure 2. Let $L_1 = \{d_1, d_2\}$ and we classify Symphony Number 9 (Opus 125) by Beethoven (d_3) in figure 3. Here we assume d_3 contains monophonic melody. We translate the first 8 or 9 bars of these music into `abc` format[13] as follows:

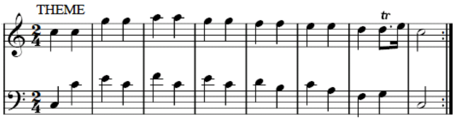

Fig. 2. Mozart Variation KV.265: *Ah, Vous dirai-Je, Maman.*

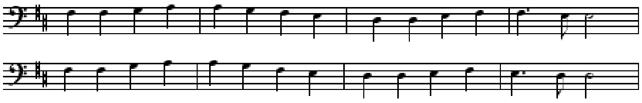

Fig. 3. Beethoven: Op.125.

```
d1: {CDEF,-}, {EDC,CDEF}, {EFGA,EDC}, {GFE,EFGA}, {CC,GFE}, {CC,CC}, {C/2C/2D/
    2D/2E/2E/2F/2F/2,CC}, {EDC,C/2C/2D/2D/2E/2E/2F/2F/2}, {CDE}

d2: {CCGG,CCEC}, {AAGG,FCEC}, {FFEE,DBCA}, {DD3/4E/4C2,FGC}, {CCGG,CCEC},
    {AAGG,FCEC}, {FFEE,DBCA}, {DD3/4E/4C2,FGC}

d3: {FFGA}, {AGFE}, {DDEF}, {F3/2E/2E2}, {FFGA}, {AGFE}, {DDEF}, {E3/2D/2D2}
```

The sequences of the pitch spectrums constitute the new features for all the bars by the tones. They are *observation sequences* (outputs) as in table 1. In this work, we consider well-tempered chords as states. Then estimation of state transition by looking at observation sequences means *chords progression*, i.e., how chords go along with polyphonic music, based on HMM. To training data, we give the states initially by hands according to the theory of music as illustrated in table 2.

It is possible to count how many times state transition arises between two states and how many output sequences are observed at each state. Eventually

Bar	d_1	d_2	d_3
1	{C:1, D:1, E:1, F:1}	{C:5, E:1, G:2}	{F:2, G:1, A:1}
2	{C:2, D:2, E:2, F:1}	{C:2, E:1, F:1, G:2, A:2}	{E:1, F:1, G:1, A:1}
3	{C:1, D:1, E:2, F:1, G:1, A:1}	{C:1, D:1, E:2, F:2, A:1, B:1}	{D:2, E:1, F:1}
4	{E:2, F:2, G:2, A:1}	{C:13/4, D:7/4, E:1/4, F:1, G:1}	{E:5/2 , F:3/2}
5	{C:2, E:1, F:1, G:1}	{C:5, E:1, G:2}	{F:2, G:1, A:1}
6	{C:4}	{C:2, E:1, F:1, G:2, A:2}	{E:1, F:1, G:1, A:1}
7	{C:3, D:1, E:1, F:1}	{C:1, D:1, E:2, F:2, A:1, B:1}	{D:2, E:1, F:1}
8	{C:2, D:2, E:2, F:1}	{C:13/4, D:7/4, E:1/4, F:1, G:1}	{D:5/2, E:3/2}
9	{C:1, D:1, E:1 }		

Bar	1	2	3	4	5	6	7	8	9
d_1	DEF	CDE	EGA	EFG	CFG	C	CEF	CDE	CDE
d_2	C	CEG	CFA	CEG	DFB	CEA	DFG	C	
d_3	FGA	FGA	DEF	EF	FGA	FGA	DEF	EF	

Table 1. Tones Appeared and Observation Sequences.

Bar	1	2	3	4	5	6	7	8	9
d_1	C	C	C	C	C	C	C	C	C
d_2	C	C	F	C	G	C	G	C	

Table 2. State Sequences by Well-Tempered Chord.

we obtain state transition diagrams with probabilities as shown in figures 4(a) (A Song of Frogs) and 4(b) (KV.265). By using the state transition diagrams, we classify d_3. Here we obtain each class membership probability by multiplying each probability of state transition and output in the corresponding diagram, but let us note we should examine extended probabilities at every state. Then we can apply MLE to d_3 through Forward algorithm or Viterbi algorithm.

Algorithm	$P(A\|d_3)$	$P(B\|d_3)$
Forward	4.006×10^{-3}	2.216×10^{-3}
Viterbi	4.006×10^{-3}	1.089×10^{-3}

In any cases, the probability A for d_3 is bigger than B and we should assign Beethoven Op.125 (d_3) to "A Song of Frogs" (A).

6 Experimental Results

6.1 Preliminaries

We examine three kinds *variations for piano* where a variation consists of themes and its various transformations. We assume all the training data are variations and considered their themes as labels.

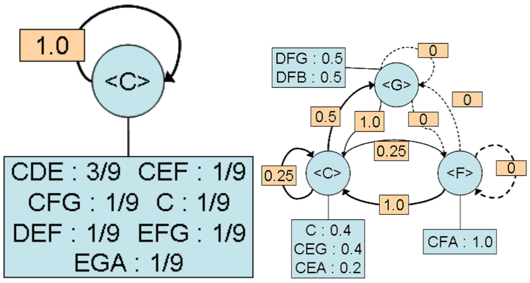

Fig. 4. A Song of Frogs (a) and KV.265 (b) - State Transition Diagrams.

Here we examine 3 variations, "Ah, Vous dirai-Je, Maman" in C Major (KV. 265) by Mozart, "Impromptus" in B flat Major (Op.142-3) by Schubert and "6 Variations on theme of Turkish March" in D Major (Op.76) by Beethoven. They contain 12, 5 and 6 variations respectively and 23 variations in total. Note there is no test collection reported so far.

All of 3 themes and 23 variations are processed in advance into a set of feature descriptions. We preprocess the feature descriptions of all the bars of the 3 themes to obtain Hidden Markov Models. Then we calculate the two collections of the feature descriptions, one for the first 4 bar, another for the first 8 bars to all the variations. In this experiment, we examine 3 kinds of chords consisting of the 3, 4 and 5 longest tones. Thus we have $23 \times 2 \times 3 = 138$ features.

Also we give a well-tempered chord to each bar of each training music by hand in advance. Once we complete all the preparations, we examine unlabeled music and guess states (chords) by our approach.

We have 3 classes (labels), *Mozart, Schubert* and *Beethoven* according to the themes. We classify all the 23 unlabeled melodies into one of the 3 labels. We say a variation is *correctly* classified if the music is composed by the label person. Formally the correctness ratio is defined as $p/23$ where p means the number of correctly classified melodies. During classification process, we also estimate a well-tempered chord to each bar, since a state in HMM corresponds to a chord (one chord in each bar).

6.2 Results

Let us show the correctness ratio to each theme in table 3 (a), and the correctness ratio to chords (in each theme) estimated by Viterbi algorithm in table 3 (b). In the latter case, we examine, to all of 8 bars, whether the chords are estimated correctly or not.

Looking at the experimental results in tables 3 (a) and 3 (b), the readers can see the perfect results (100.0 % of the correctness ratio) for theme classification with *only* 3 tones of every case of bars. Similarly we get 69.6 % of the correctness ratio (chord) with 3 tones in 4 bars or more.

	Algorithm	
(tone) bars	Chord(Forward)	(Viterbi)
(3) 4	100.0%	95.7%
(3) 8	100.0%	100.0%
(4) 4	69.6%	56.5%
(4) 8	82.6%	78.3%
(5) 4	39.1%	47.8%
(5) 8	30.4%	47.8%

	Number of Correct Bars		
tone	8bars	$\geq$ 6bars	$\geq$ 4 bars
3	4.35%	39.1%	69.6%
4	0.0%	38.9%	55.6%
5	0.0%	45.5%	63.6%

Table 3. Correctness Ratio - (a) Theme and (b) Chord.

6.3 Discussion

Let us look closer at our results. As for the correctness ratio of theme classification in table 3, we see the more bars cause better results. In fact, in a case of 3 tones by Viterbi algorithm, we have improved the results of 95.7% with 4 bars to 100.0% with 8 bars. Similarly in a case of 4 tones by Forward algorithm, we have improved 69.6% with 4 bars to 82.6% with 8 bars, and 56.7% with 4 bars by Viterbi algorithm to 78.3% with 8 bars. However, in a case of 5 tones, we got the worse result of 39.1% with 4 bars by Viterbi algorithm to 30.4% with 8 bars. One of the reasons is that, the more bars we have, the more chances we get to make mistakes in a case of two similar themes. We have already pointed out this problem in another work[15].

In this experiment, we have compared Viterbi and Forward algorithms with each other. In cases of correctness ratio of theme classification with respect to 3 tones and 4 tones, Forward algorithm is superior to Viterbi, while in a case of 5 tones, Viterbi is better. One of the specific points to Forward algorithm is that, the more bars we give, the worse the classification goes, but not in a case of Viterbi algorithm. This comes from the difference of probability calculation, though we skip the detail of probability results. In Forward algorithm, we get the probability by summarizing the values along with all the paths to the state of interests. On the other hand, in a case of Viterbi, we get the probability by finding one of the paths with the highest probability.

There is no investigation of polyphony classification to compare directly with our results. In [9, 10], given about 3000 music of polyphony, classification as been considered as query and the results have been evaluated based precision. They got 59.0% at best.

Let us examine our previous results[14]. Naive Bayesian provides us with 87.0 % correctness ratio, and we got 91.3 % correctness ration at best by EM algorithm. Compared to our case where all the melodies are polyphony that are much complicated, we got the perfect correctness ratio (100.0 %) with 8 bars. Certainly our approach is promising.

As for chord estimation, we got up to 70% correctness ratio with respect to 4 bars and more cases. Basically it is possible to say that HMM approach works well.

7 Conclusion

In this work, we have proposed a sophisticated approach to classify polyphonic melodies given a small amount of training polyphonic music. To do that, we have introduced special features of pitch spectrum and estimated Maximum Likelihood based on HMM. We have shown the perfect estimation of theme classification by examining small amount of tones and bars in unknown melodies.

References

1. DowlingW.J.: Scale and Contour – two components of a theory of memory for melodies, *Psychological Reviews* 85-4, pp.341-354, 1978.
2. Droettboom, M. et aI.: An Approach Towards A Polyphonic Music Retrieval System, *Intn'l Symp. on Music Information Retrieval* (ISMIR), 2001.
3. Droettboom, M. et aI.: Expressive and Efficient Retrieval of Symbolic Musical Data, *Intn'l Symp. on Music Information Retrieval* (ISMIR), 2002.
4. Grossman, D., O'.Frieder: Information Retrieval – Algorithms and Heuristics, Kluwer Academic Press, 1998.
5. Ishigeta, M. et al: Theory of Music, Ongaku-No-Tomo-Sha, 2001 (in Japanese).
6. Iwasaki, M.: Foundation of Incomplete Data Analysis, Economicst Sha, 2002 (in Japanese).
7. Kim, Y. et al.: Analysis of A Contour-based Representation for Melody, *Intn'l Symp. on Music Information Retrieval* (ISMIR), 2000.
8. Miura, T. and Shioya, I.: Similarities among Melodies for Music Information Retrieval, ACM *Conf. on Information and Knowledge Management*(CIKM), 2003.
9. Pickens, J.: A Comparison of Language Modelling and Probabilistic Text Information Retrieval Approaches to Monophonic Music Retrieval, *Intn'l Symp. on Music Information Retrieval* (ISMIR), 2000.
10. Pickens, J. and Crawford, T.: Harmonic Models for Polyphonic Music Retrieval, ACM *Conf. on Information and Knowledge Management*(CIKM), 2002.
11. Uitdenbogerd, A.L. et al.: Manipulation of Music For Melody Matching, ACM *MultiMedia* Conf., 1998.
12. Uitdenbogerd, A.L. et al.: Music Ranking Techniques Evaluated, *Intn'l Symp. on Music Information Retrieval* (ISMIR), 2000.
13. Walshaw, C.: abc Version 1.6, `www.gre.ac.uk/~c.walshaw/abc2mtex/abc.txt`.
14. Yoshihara,Y. and Miura, T: Melody Classification Using EM Algorithm. *Computer Software and Applications Conference*(COMPSAC), pp. 204-210, 2005.
15. Yoshihara,Y. and Miura, T: Classifying Polyphonic Music Based on Markov Model. *Intelligent Data Engineering and Automated Learning*(IDEAL), pp. 697-706, 2006.