Expert Systems with Applications 36 (2009) 7666–7674

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Emotion-based music recommendation by affinity discovery from film music ☆

Man-Kwan Shan [a,*], Fang-Fei Kuo [b], Meng-Fen Chiang [a], Suh-Yin Lee [b]

[a] Department of Computer Science, National Chengchi University, Taiwan 11605, Taiwan
[b] Department of Computer Science, National Chiao-Tung University, Taiwan

ARTICLE INFO

Keywords:
Music recommendation
Emotion detection
Affinity discovery

ABSTRACT

With the growth of digital music, the development of music recommendation is helpful for users to pick desirable music pieces from a huge repository of music. The existing music recommendation approaches are based on a user's preference on music. However, sometimes, it might better meet users' requirement to recommend music pieces according to emotions. In this paper, we propose a novel framework for emotion-based music recommendation. The core of the recommendation framework is the construction of the music emotion model by affinity discovery from film music, which plays an important role in conveying emotions in film. We investigate the music feature extraction and propose the Music Affinity Graph and Music Affinity Graph-Plus algorithms for the construction of music emotion model. Experimental result shows the proposed emotion-based music recommendation achieves 85% accuracy in average.

1. Introduction

Digital music has become popular in human life, owing to the advancement of the digital music technology. The rapid growing demand for music management techniques and digital music applications makes music information retrieval an important research field. The main goal of music information retrieval is to retrieve a specific (set of) music object and it requires users providing information about the music to be retrieved, such as title, lyrics or humming tune. An important branch of music information retrieval research is personalized music recommendation. Personalized music recommendation techniques attempt to filter out the music a user dislikes and recommend that a user might like. The techniques typically make recommendation by analyzing a user's preference via music accessing behavior, rather than user-specified preference information. There exist three major approaches for the personalized music recommendation. One is the content-based filtering approach which analyzes content of the music which users liked in the past and recommends similar music (Kuo & Shan, 2002). Another is the collaborative filtering approach which recommends music that peer group of similar preference liked (Shardanand & Maes, 1995). The other is the hybrid approach which integrates the content and collaborative information for personalized music recommendation (Chen & Chen, 2001; Yoshii, Goto, Komatani, Ogata, & Okuno, 2006).

These recommendation approaches are based on the users' preferences observed from the listening behavior. However, sometimes, the music a user needs is decided by the emotion of the user or context. Most people experience music every day with affective response. For example, we may feel cheerful when listening to an excellent performance at a concert, and may feel sad when listening to the music of a late night movie. Consequently, recommending music according to emotion better meets the users' requirement in some cases.

Some researchers have devoted to the understanding of the relationships between music and emotion from the philosophical, musicological, psychological and anthropological perspectives (Gabrielsson & Lindstrom, 2001; Tao & Ogihara, 2004). To recommend music based on emotions, the straightforward approach is to recommend music by the rules, in terms of the relationship between emotion and music elements, observed by the psychological research. Another possible approach is to learn the rules by training from music pre-labeled with emotion types. However, the emotion labeling is time-consuming.

In our work, we propose a generic framework for emotion-based music recommendation by affinity discovery from film music. In particular, we investigate music feature extraction and propose a modified Mixed Media Graph (MMG) algorithm, Music Affinity Graph (MAG), to discover the relationship between music features and emotions from film music. However, both MAG and MMG algorithms have the problem that the *discrimination powers* of the discovered features are not necessarily high. It means that the discovered features might be highly related to not only the query emotions but also other emotions. Consequently, we propose the Music Affinity Graph-Plus (MAG-Plus) algorithm to take the discrimination power into consideration. We also discuss some

☆ Part of the content of this paper has been published in ACM Proceedings of International Conference on Multimedia, 2005.

* Corresponding author. Tel.: +886 2 29393091x67622; fax: +886 2 22341494.
E-mail address: mkshan@cs.nccu.edu.tw (M.-K. Shan).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.09.042

existing researches on film emotion detection, which can be used in the recommendation framework to avoid labor work in emotion labeling. Potential applications of our proposed emotion-based music recommendation framework include music therapy, music score selection for production of home video, background music playing in shopping mall to stimulate sales, and music playing in context-aware home to accommodate inhabitants' emotion.

2. The proposed emotion-based music recommendation framework

2.1. Framework overview

Fig. 1 shows the process of the proposed generic music recommendation framework. The heart of the framework is the construction of the music emotion model from film music, owing to the close relation between the emotion and music in films. Kalinak (1992) has claimed that music is "the most efficient code" for emotional expression in film. A film music composer usually composes music according to a scenario. The purpose of the composition generally agrees with how audiences react to it. The major roles of film music include: (1) the overture for suggesting the theme or spirit of the whole film; (2) the expression of emotions, thoughts, wishes and characterizations of the characters; (3) to change the audiences' emotions and to be used as the prophetic sign; (4) to suggest situations, classes or ethnic groups; (5) to neutralize or even reverse the predominant mood of a scene (Giannetti, 2004). Consequently, we think film music is a rich, explicit and direct source of the emotional music and is suitable for music emotion model construction.

The music emotion model is constructed by the following steps:

1. Film music emotion detection: A piece of film music corresponds to a film segment. In most cases, the emotion of the film music accords with that of the plot or circumstance of the corresponding film segment. Consequently, the film music emotion can be detected from film video, which provides useful cues, such as caption, speech, sound effect, and visual features. The film segment emotion detection can be done by multi-modal approach using the above-mentioned cues. Both results of emotion detection and feature extraction for film music can be stored in the repository for later use. Current researches on emotion detection of films are reviewed in Section 2.2.
2. Film music feature extraction: Some music features are extracted from film music to represent the emotional characteristics. In our work, the types of features include mode, melody, rhythm and tempo. Section 3 will present the details of film music feature extraction.
3. Affinity discovery: The last step of music emotion model construction is to discover the affinities between the film music features and emotions. The discovered affinities can be used to find which feature types and values determine some emotions. Section 4 will describe the proposed affinity discovery approach.

The music pieces to be recommended to listeners are selected from a music database. Unlike the music used for music emotion model construction, the database music does not limit to film music. The same music features are extracted from database music and stored for the recommendation. Given the query emotions (i.e., the set of emotions according to a listener's need), the music emotion model will return the recommended music features with respect to the query. The recommended features are then employed to rank the database music and to recommend music for query emotions.

2.2. Film emotion detection

Several studies have been done on film emotion detection from text, visual content, and sound. Moncrieff, Dorai, and Venkatesh (2001) proposed an algorithm for affective sound event detection through sound energy dynamics of films. Four types of sound energy events are identified based on attack, sustain, and decay of sound. Adams, Dorai, and Venkatesh (2002) utilized the attributes of motion and shot length to measure tempo of a movie. Extraction of expressive sections and events is achieved by edge detection of tempo flow plots. The research by Salway and Graham (2003) extracts emotions from audio description scripts in films provided for visually impaired people. A list of emotion tokens was generated using WordNet for each of the selected 22 types of emotions. Occurrences of emotion tokens are the indication of emotion being depicted. Kang (2003) utilized Hidden Markov Model to detect affective events, such as fear, sadness, and joy. For each shot, color histogram, motion intensity and shot cut rate are extracted and transformed to observation vector sequences. Then the observation vector sequences are decoded into the most likely sequence of Hidden Markov Model. Based on film theories and psychological models, Wei, Dimitrova, and Chang (2004) proposed two color representations, movie palette histogram and mood dynamic histogram, for color-mood analysis of films. Movie palette histogram is a global measure for the color palette while mood dynamic histogram is a discriminative measure for the transitions of the moods. Along with the dominant color ratio and the pace of the movie, these color representations are fed into Support Vector Machine for classification of eight types of moods, anger, fear, joy, sorrow, acceptance, rejection, surprise, and expectancy. Hanjalic and

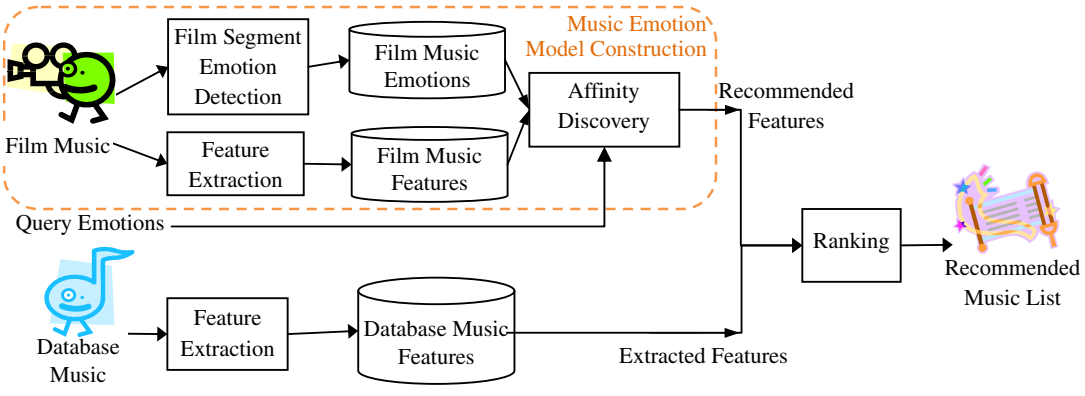

Fig. 1. The proposed music recommendation framework.

Xu (2005) exploited motion activity, cut density and sound energy to map the affective video content to the dimensional (arousal-valence) emotion space which is one of the popular emotion model developed in the Psychology field (Ortony et al., 1988). By computing and integrating the arousal and valence curve for a video, the so-called "affect curve" is generated to represent the emotion transition along the video. The study by Xu, Chia, and Jin (2005) extracts the affective contents of comedy and horror films by detecting a set of audio emotional events such as laughing, horror. The acoustic features are extracted from audio and the observation vectors are generated for audio event classification based on Hidden Markov Model. In the approach proposed by Chan and Jones (2005), the arousal and valence features extracted from film audio frame are mapped onto a set of emotional keywords. To label the dominate emotions in a film segment, clusters of emotion keywords are located in a sequence of frames.

All these studies may be utilized to help detecting emotions automatically from captions, visual features of scenes or acoustic features of dialogs in films. In this paper, we do not address the issues of film emotion detection. We assume that the emotions associated with film music have been detected.

3. Music feature extraction

Music elements which affect the emotion include melody, rhythm, tempo, mode, key, harmony, dynamics and tone-color. Among these music elements, melody, mode, tempo and rhythm have stronger effects on emotions. Generally speaking, major scale is brighter, happier than minor; rapid tempo is more exciting or tenser than slow tempo.

Take Schubert's *Der Lindenbaum* from *Winterreise* cycle as an example. It describes a man who is disappointed in love and drifts from home recalls the linden tree at home. First, Schubert used *E* major scale to express memory of the warm past at home (Fig. 2a). Then, it is modulated from *E* major to *E* minor to reflect the mournful situation of the wanderer (Fig. 2b).

An example for tempo's effect is Saint–Saens' *Tortoises* from *Carnival of the Animals* (Fig. 3). Saint–Saens quoted the main melody of *Orpheus in the Underworld Overture* (*Can–Can*) by Jacques Offenbach, which is brisk, joyful dance music. The tempo of melody in *Tortoises* is very slow to represent this steady animal.

Sometimes emotion conveyed by music cannot be identified using only one of the above elements. For example, the music in minor scale but sprightly rhythm may be joyful rather than sad. Consequently, we investigate the effect of the combination of three types of features and propose the corresponding feature extraction algorithms related to these music elements.

3.1. Mode and melody

In music perception, mode is the most important feature which determines the emotion. As the above-mentioned example shown in Fig. 2, the change in mode can result in the opposite emotion. Another important feature is melody, which is the most memorable element in music and correlates closely with mode. Pitch, interval contour and average pitch value are common-used melody features. However, these melody features may not reflect the emotion expressed by the music. For instance, in Fig. 2, melody lines in two excerpts are similar while the expressed emotions are quite different.

In our previous work on music style recommendation, we utilized chord as the melody feature for representing the music style and also proposed an algorithm for assigning chords for melody (Kuo & Shan, 2002). In this work, to aim at the music emotion, we proposed an algorithm to identify the mode and key of a music piece and further combined the mode and melody features. We modified the chord assignment algorithm to take mode and key of melody into consideration. Consequently, the assigned chords imply the information about melody, mode and harmony, which are influential elements for music emotion.

To extract the mode/key and assign the chords, the original polyphonic music should be pre-processed to obtain the main melody sequence. Some melody extraction algorithm can be used for MIDI files, such as all-mono (Uitdenbogerd & Zobel, 1999). MIDI key signature event includes the information of both mode and number of sharps or flats. However, many MIDI files do not keep the mode information. In these files, mode is always major whether the real one is major or not. In our MIDI database, almost all files which we collected from Internet omit the mode information. To identify real mode and key from MIDI files, we proposed the key signature identification algorithm.

The key signature identification algorithm utilizes music theory and some heuristic rules to identify the mode, then determines key signature by mode and number of sharps/flats. Given number of sharps or flats, the key signature has two possibilities, shown in Table 1. In Table 1, the first row is the number of sharps/flats (positive for sharps and negative for flats.) For instance, the key signature with no sharps and flats (0) may be C major or A minor and that with two flats (−2) may be Bb major or G minor.

The key signature identification algorithm is shown in Fig. 4. Music is first divided into several parts; there is no change of number of sharp/flat in each part (Line 1). For each part, we can find two possible key signatures from Table 1. Then, we check appearance of the leading tone of the minor key. Leading tone is a half step lower than the tonic note and as its name indicates, it is

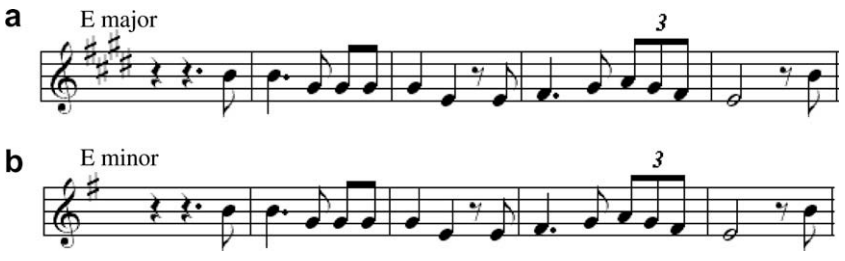

Fig. 2. Excerpts of *Der Lindenbaum*, in Schubert's *Winterreise*.

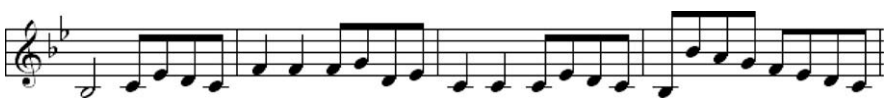

Fig. 3. Main theme of *Tortoises* by Saint–Saens and *Can–Can*, in Offenbach's *Orpheus in the Underworld*.

Table 1
Key signatures table

Number of sharps/flats	−7	−6	−5	−4	−3	−2	−1	0	1	2	3	4	5	6	7
Major	Cb	Gb	Db	Ab	Eb	Bb	F	C	G	D	A	E	B	F#	C#
Minor	Ab	Eb	Bb	F	C	G	D	A	E	B	F#	C#	G#	D#	A#

Algorithm *Key_Signature_Identifcation*

Input: music object m, threshold t

Output: sequence of key signatures

1. if m has modulations, divide m into parts
2. for each part p_i
3. find possible key signatures k_{major} and k_{minor} from Table
4. compute leading tone of k_{minor}
5. compute the frequency of appearance f of leading tone
6. if $f > t$ then $K_i = k_{minor}$
7. else $K_i = k_{major}$
8. return all K_i

Fig. 4. Key signature identification algorithm.

strongly leading to tonic note. Consequently, leading tone is the most important note which defines the key signature. Moreover, leading tone of minor scale is not one of the scale notes. For example, the leading tone of C minor is B, which is not one of the scale notes C, D, Eb, F, G, Ab, and Bb. The notes which are not the scale notes appear less than scale notes. This means that if the appearance frequency of leading note is high enough (larger than user-defined threshold t), it is likely to be minor scale.

Our modified chord assignment algorithm is a heuristic method based on the music theory and Harmony. The algorithm selects suitable chords from the candidates according to consonance and chord progression. The candidate chords considered here are the diatonic triads, which are basic and common chords. The diatonic triads of natural and harmonic minor scales are a little different. Consequently, we select 9 diatonic chords which are often used in composition from both natural and harmonic minor scales. Fig. 5 shows two sets of the candidate chords in C major and C minor Table 2.

Before assigning chords to a music object, the melody should be divided into segments according to the density of notes in the music object. A segment will be assigned a chord (or a set of chords, if the most suitable chords are not unique). The music segmentation algorithm is shown in Fig. 6.

Then, the chord assignment algorithm is applied to each music segment respectively. The chord assignment algorithm consists of two stages. In the first stage (Lines 1–13 in Fig. 7), the strategy for scoring the candidates is as follows:

1. If the candidate has more notes which also appear in the segment, it gets more points.
2. The longest note should be more dominant in the segment; therefore, candidates that have the longest note get points.
3. Tonic triad (*I* for major and *i* for minor) get more points in the first and the last segments, because music often begins and ends at tonic triad.

For each segment, if the highest-score candidate is not unique, proceed to the second stage. In the second stage (Lines 14–21 in Fig. 7), rules of chord progression which includes *root motion* and *dissonance resolving* are used.

1. Root motion: Root motion means the movement from one chord's root note (i.e., lowest note) to next chord's root note. We selected some common root motions for scoring, such as down a fifth (the roots of the adjacent two chords move down by the interval a fifth, ex. I → IV) or up a second (IV → V).
2. Dissonance resolving: Some chords are unstable and tend to resolve to more stable chords such as tonic triad. Therefore, if a chord in the previous part is unstable, some candidates that are more stable will get points.
3. Finally, if the highest-score candidate is not unique, we assign a set of these candidates for each segment, named as the chord-set.

3.2. Rhythm and tempo

Rhythm is the music feature that describes the timing information of music. Our rhythm extraction method includes the following steps: First, a basic time unit is decided by the duration of the shortest note, and the beat sequence is extracted based on percussion instruments. The beat sequence is represented as a binary string where one stands for the onset of a percussion note and the following zero(s) stand for the duration of that note. For instance, a quarter notes can be represented as the binary string 1000, where the basic unit is set to sixteenth note long. Second, the repeating patterns are discovered from the beat sequence using existing repeating pattern finding algorithm (Hsu, Liu, & Chen, 1998). The rhythmic pattern of music is the recurrent pattern with high frequency. We retained the highest frequency patterns for the music.

In our approach, tempo is calculated from resolution of the music and beat density of the most repetitive pattern. The resolution of a music object is the number of ticks per beat. The following is the formula for calculating tempo:

$$\text{tempo} = \text{resolution} * \text{NB}/\text{NS}, \quad (1)$$

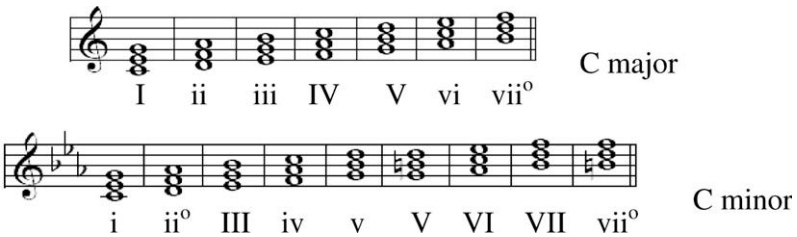

Fig. 5. Candidate chords in C major and C minor.

Algorithm *Music-Segmentation*

Input: music object m

Output: sequence of music segments MS

1. if m has changes in time signature, divide m into parts. Each part has one time signature.
2. for each part p_i
3. for each type t of notes in p_i, count appearance frequency f_t($t \subset$ *fourth-note, eighth-note, and etc.)*
4. if $\max_t f_t = f_{sixteenth-note}$ and $f_{sixteenth-note} \geq 2 f_{eighth-note}$
   then divide p_i into half-measure length segments
5. else divide p_i into one-measure length segments
6. return all music segments MS

Fig. 6. Music segmentation algorithm.

Algorithm *Chord-Assignment*

Input: previous chord-set *pre_c,* music segment *ms (ms* $\in$ *MS),* key signature *k,* set of chord candidates C_{major}, C_{minor}

Output: chord-set CS

1. if k is major, candidate set $C = C_{major}$
2. else $C = C_{minor}$
3. for each candidate chord ch, initialize the score S_{ch} to zero
4. if ms is the first or the last segment of m
   then score of chord I , S_I += 10
5. for each distinct pitch p, accumulate the duration $D(p)$
6. $\mathbf{P} = \{\text{all longest pitches } p\}$
7. if $\max D(p) \geq 0.5 \times D(ms)$ then $score = 2$
8. else if $|\mathbf{P}| = 1$ then $score = 1$ else $score = 0$
9. for each chord ch do
10. for each distinct pitch p in ms do
11. if *ch contains p* then S_{ch} ++
12. if $ch \cap \mathbf{P} \neq \phi$ then S_{ch} += $score$
13. if cardinality of $\{ch \mid S_{ch} = \max_i S_i\} = 1$ then return ch
14. else if ms isn't a part of the first measure of m do
15. for each chord ch do
16. if *root(pre_c)* != leading note and *root(ch)* is descending 5th , descending 3rd or descending 4th of *root(pre_c)* then S_{ch} += 2
17. if *root(pre_c)* is subdominant, dominant or leading note and *root(ch)* is ascending 2nd then S_{ch} += 2
18. if *cardinality* of $\{ch \mid S_{ch} = \max_i S_i\} = 1$ then return ch
19. else for each chord ch do
20. if *root(ch)* = lowest pitch in ms then S_{ch} += 2
21. return $CS = \{ch \mid S_{ch} = \max_i S_i\}$

Fig. 7. Chord assignment algorithm.

where NB is number of beat onset in a rhythmic pattern and NS is the length of the rhythmic pattern.

4. Affinity discovery and music recommendation

Emotion-based music recommendation recommends several music pieces corresponding to the query emotions. More precisely, given a query set of emotions, we wish to find out the corresponding music features to rank the database music. The affinities between music features and emotions should be discovered from training data. The affinity graph algorithm, Mixed Media Graph, is adopted and modified for the proposed emotion-based music recommendation.

MMG was proposed to find correlations across the media in a collection of multimedia objects (Pan, Yang, Faloutsos, & Duygulu, 2004). A typical application of MMG is the automatic image captioning to automatically assign caption words to the query image. This is achieved by finding correlations between the image features and the caption words from a given collection of images and associated captions.

In MMG graph, all the objects and associated attributes are represented as vertices. For objects with n types of attributes, MMG graph will be an $(n + 1)$ layered graph with n types of vertices plus one more type of vertices for the objects. There are two types of edges in MMG graph. The object-attribute-value link (OAV-link) is the edge between an object vertex and an attribute vertex. The other type, nearest neighbor link (NN-link), is the edge between two attribute vertices. An edge is constructed between each attribute vertex and each of its k nearest neighbors. After the construction of MMG graph, to find the correlations across the media, the

mechanism of random walk with restart is employed to estimate the affinity of attribute vertices with respect to the query vertices. In detail, $a_q(v)$, the affinity of vertex v with respect to query vertex q is the steady-state probability that the random walker will reach v from q. In each vertex, the walker randomly selects and moves to the next vertex among the available edges with the exception to return to q with restarting probability c.

Let N be the number of nodes in the MMG graph. The steady-state probability vector, $\vec{\mathbf{a}}_q = (a_q(v_1), \ldots, a_q(v_N))$, is estimated by the following equation:

$$\vec{\mathbf{a}}_q = (1-c)\boldsymbol{A}\vec{\mathbf{a}}_q + c\vec{\mathbf{r}}_q, \tag{2}$$

where $\boldsymbol{A}$ is the column-normalized adjacency matrix of the MMG graph and $\vec{\mathbf{r}}_q$ is the restart vector, which is a column vector with all entries zero excluding the entry corresponds to q. $\vec{\mathbf{a}}_q$ is initialized to $\vec{\mathbf{r}}_q$ and computed iteratively until convergence is achieved.

Example For the sake of illustrating MMG algorithm, let's take a simplified example from Fig. 9a. Fig. 9a shows a MMG graph consist of the object set $\{Q_1, Q_2\}$, query object O_q and two types of attribute vertices $\{at_1, at_2, at_3\}$ and $\{x, y\}$. The solid lines represent the OAV-links and the dotted lines indicate the NN-links. The number beside each node indicates the corresponding index in the adjacency matrix $\boldsymbol{A}$, the steady-state probability vector $\vec{\mathbf{a}}_q$ and the restart vector $\vec{\mathbf{r}}_q$. Suppose the number of nearest neighbors k is one and the restarting probability c *is* 0.2. The adjacency matrix A of the graph is shown in Fig. 8b and the initialized $\vec{\mathbf{a}}_q$ is initialized to the restart vector, $(1,0,0,0,0,0,0,0)^T$.

Fig. 9 illustrates the process of computing the steady-state probability vector $\vec{\mathbf{a}}_q$. The numbers beside nodes shows the value of $\vec{\mathbf{a}}_q$ in the current iteration. As the figure shows, the probability spreads from the query object node to neighbor nodes. $\vec{\mathbf{a}}_q$ converges after seven iterations and it indicates that y has closer affinity to O_q.

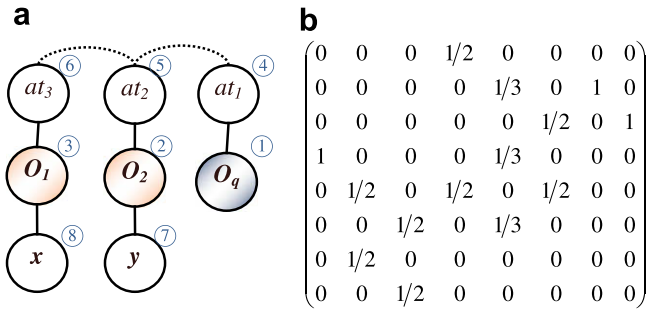

Fig. 8. (a) Example of the mixed media graph. (b) The adjacency matrix A for (a).

4.1. Music Affinity Graph algorithm (MAG)

The Music Affinity Graph is constructed as follows. For each trained music object, a music object vertex is created. Four types of attribute vertices – emotion, melody, rhythm, and tempo vertices are created and attached to each music object vertex. For the emotion, rhythm and tempo vertices, each vertex corresponds to one instance. Vertex creation for the melody feature, which is represented as a set of chord-set, is slightly different. For each chord-set, one vertex is created. In other words, for a music object in which the melody feature is represented as a set of n chord-sets, there are n melody vertices associated with the corresponding music object vertex.

The edge between every two melody (or tempo) vertices is constructed based on the k-nearest neighboring while the edge between every two emotion (or rhythm) vertices is constructed only when both vertices are of the same emotion (or rhythm). The steady-state probability vector for the Music Affinity Graph is also computed by Eq. (2).

Example Given a collection of two music objects $\{m_1, m_2\}$ in which music object m_1, with melody feature $\{c_{11}\}$, tempo feature t_1, rhythm feature r_1, is perceived with emotions $\{e_A, e_B, e_C\}$ while music object m_2, with melody feature $\{c_{21}, c_{22}\}$, tempo feature t_2, rhythm feature r_2, is perceived with emotions $\{e_A, e_B\}$. Fig. 10 illustrates the constructed Music Affinity Graph with respect to the query music Q and query emotions $\{e_A, e_B\}$ where the number of nearest neighbors, k, is set to one.

4.2. Music Affinity Graph-Plus algorithm (MAG-Plus)

The performance of the Music Affinity Graph may be improved by the consideration of *discrimination power of the music feature values*. The steady-state probability of an attribute vertex represents the affinity between the corresponding music feature value and the query emotions. However, we may not determine a music piece which has the high-affinity features belongs to the query emotions. It is possible that the music feature value has high affinity with respect to both query emotions and other non-query emo-

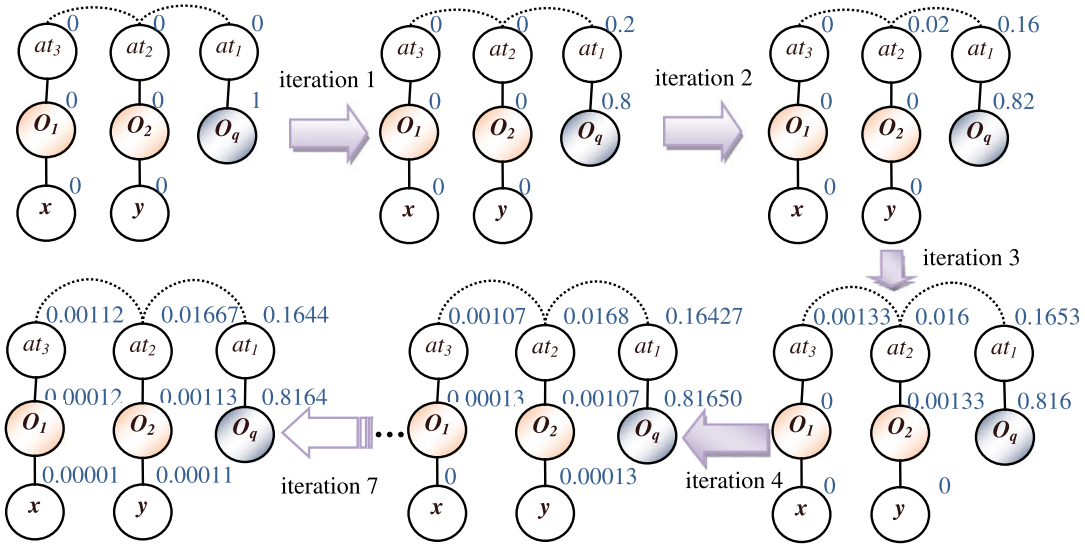

Fig. 9. Illustration of the steady-state probability estimation from Fig. 8a.

tions. In other words, different music emotions might have the same music feature values, and the discrimination powers of those feature values are low. For example, we cannot use a chord-set to distinguish happy from sad music if it appears in both happy and sad music pieces. To address this problem, we proposed the *complement affinity graph G′*, which is the modification of the Music Affinity Graph. The complement affinity graph G' is similar to the Music Affinity Graph G except that all the nearest neighbor links from the query emotions are removed while the complement query edges are added. The complement query edge connects the query emotion vertex to the vertex, associated with music, of different emotion. Fig. 11 shows the complement affinity graph of Fig. 10. The original links from query emotions (edge E_1, E_2, E_3 and E_4 in Fig. 10) are removed, and the complement query edge (the dashed lines) are added in Fig. 11. After the construction of complement affinity graph, using the mechanism of random walk with restart, the affinity $g_q(v)$ for each vertex v in G' can be derived in the same manner of the affinity $f_q(v)$ in Music Affinity Graph G. Consequently, the final affinity $h_q(v)$ is equal to $f_q(v) - g_q(v)$.

4.3. Music ranking

Once the affinity $h_q(v)$ is derived, a ranked music list could be generated according to the affinity $h_q(v)$. The affinity values of the music feature vertices indicate the degrees of relationship between the feature vertices and query emotions. Consequently, for each music feature (i.e., melody, rhythm and tempo in this work), we select top-r feature values as the recommended features. One may notice that we select top-r values for each feature, respectively rather than for entirety of all features. Let R be the set of all top-r feature values of melody, rhythm and tempo. For each database music m, the ranking score RS(m) is defined as

$$\mathrm{RS}(m) = \sum \mathrm{MA}_m(v_f), \tag{3}$$

where $\mathrm{MA}_m(v_f)$ is the matching affinity of m versus feature vertex v_f, and $v_f \in R$. $\mathrm{MA}_m(v_f)$ is defined as

$$MA_m(v_f) = \begin{cases} h_q(v_f), & \text{if } v_f \text{ satisfies } m, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

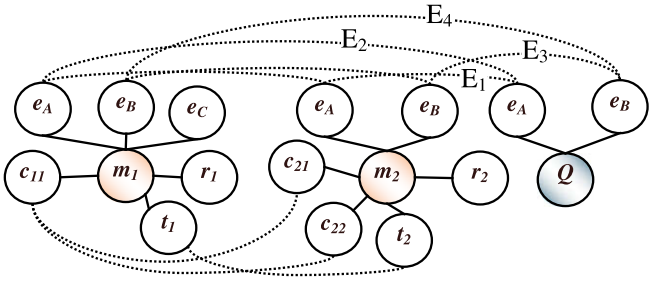

Fig. 10. The Music Affinity Graph G.

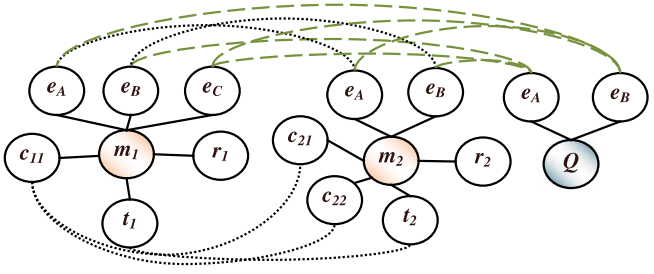

Fig. 11. The complement Music Affinity Graph G.

The returned music piece is selected by ranking score RS(m). Higher ranking score means higher possibility the music belongs to the query emotion.

5. Performance evaluation

To evaluate the effectiveness of our proposed music recommendation approach, we performed experiments on a collection of 107 film music from 20 animated films. We choose animated films because emotions in animated films are more clear and explicit in general. The 20 films include the productions of Disney, Studio Ghibli and DreamWorks, such as Lion King, Spirited Away, and Shrek. The MIDI files of Studio Ghibli were collected from the website "A Dedication to Studio Ghibli Films" (http://www.wingsee.com/ghibli/), the MIDI files of Disney's films were downloaded from website "Gineva's Disney Page" (http://www.ginevra2000.it/Disney/Midi/allmidi.htm), and the remaining ones were collected from HamieNET.com (http://www.hamienet.com).

To simplify the experiments, the emotions of the music were annotated manually. The emotions used in our experiments were mainly selected from (Reilly, 1996). We added some emotions such as lonely and nervous, and divided these emotions into 15 groups. These groups are shown in Table 2. Each MIDI file was annotated with one to seven emotion groups.

We took five-fold cross-validation in our experiments. For each test music piece, the affinity graph was constructed from the training set, and the emotions of the test music piece are used as the query emotions. In average, the Music Affinity Graph contains 1000 chord-set nodes, 180 rhythm nodes, 86 tempo nodes and total 1600 nodes. All database music objects were ranked by the approach stated in Section 4.3. Top-10 music pieces were returned.

The recommendation performance is measured by the similarity between query emotions and returned music's emotions. The performance measure used in our experiments is defined as

$$\text{average\_score} = \sum_{i=1}^{N} \text{Score}_i / N, \tag{5}$$

where N is the number of returned music pieces, Score_i is the similarity between emotion sets E_i of the ith returned music piece and E_q of the query. Score_i is defined as

$$\text{Score}_i = |E_i \cap E_q| / \sqrt{|E_i| \times |E_q|}, \tag{6}$$

where $|E_i|$ is the cardinality of the set E_i, $\cap$ is the set intersection operation. $\text{Score}_i = 1$ if E_i is the same as E_q.

Fig. 12 shows the performance of the proposed recommendation approach (with $k = 7$ and $c = 0.8$, where k is number of the

Table 2
Types of emotions used in the experiments

No.	Emotions
1	Hope, Joy, Happy, Gloating, Surprise, Excited
2	Love
3	Relief
4	Pride, Admiration
5	Gratitude
6	Gratification, Satisfaction
7	Distress, Sadness
8	Fear, Startle, Nervous
9	Pity
10	Resentment, Anger
11	Hate, Disgust
12	Disappointment, Remorse, Frustration
13	Shame, Reproach
14	Lonely
15	Anxious

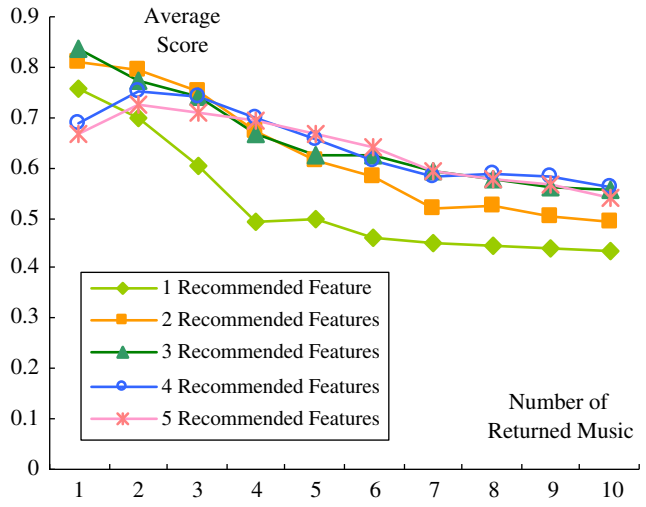

Fig. 12. Performance of MMAG-Plus algorithm with various numbers of recommended features. ($k = 7$, $c = 0.8$).

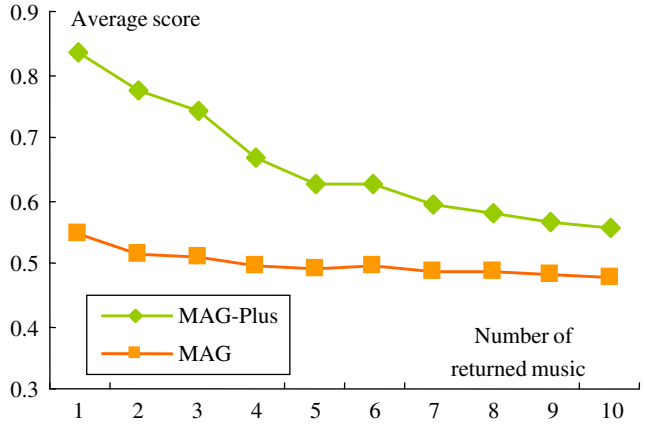

Fig. 13. Performance comparison between MAG-Plus and MAG algorithms ($k = 7$, $c = 0.8$).

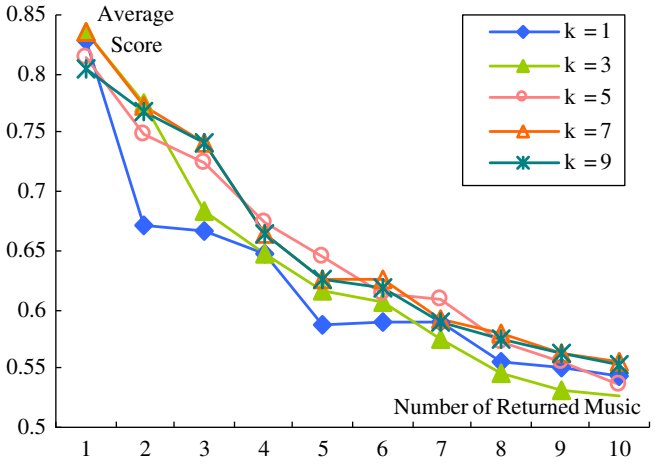

Fig. 14. Performance of MAG-Plus algorithm with various k ($c = 0.8$).

nearest neighbors of attribute vertices and c is the restart probability). The average scores of top-one music are above 0.8 using two or three recommended features. The overall average scores are above 0.5. The result using more than one recommended features is better; it is possible that information of only one recommended feature is insufficient for recommendation.

The comparison between our proposed MAG-Plus algorithm and MAG algorithm is shown in Fig. 13. The values of parameters, k and c, are the same as those of previous experiment, and the number of recommended features is 3. The figure shows that MAG-Plus outperforms MAG algorithm. In particular, MAG-Plus has remarkable improvement in top-1 to top-4 returned music, where the average scores of MAG-Plus are between 0.75 and 0.95, and those of MAG are between 0.5 and 0.55. The result shows the discrimination power of music features has a great influence on the performance of emotion-based recommendation. In our experiments, we find the number of nearest neighbor, k, has small effect on the performance. Fig. 14 shows the results of various k (with $c = 0.8$, number of recommended features is 3). We can see in this figure, the average score curves are quite similar. The only curve which is relatively unstable is $k = 1$. The reason may be little information is provided by only one nearest neighbor.

6. Conclusions

In this paper, we presented a generic framework to recommend music based on emotion. The core of our proposed recommendation framework is to construct the music emotion model from film music, for music plays an important role in conveying emotions in film. The construction process of music emotion model consists of feature extraction, emotion detection and association discovery. We proposed the feature extraction approaches to extract chord, rhythm and tempo. For the association discovery between emotions and music features, we proposed the Music Affinity Graph and the Music Affinity Graph-Plus algorithms. The MAG-Plus algorithm solves the discrimination power problem in both the MAG and Mixed Media Graph algorithm. Experimental result shows that the top-one average score of the top-one result achieves 0.85 using three recommended features.

References

Adams, B., Dorai, C., & Venkatesh, S. (2002). Toward automatic extraction of expressive elements from motion pictures: Tempo. *IEEE Transactions on Multimedia, 4*(4), 2002.

Chan, C. H. & Jones, G. J. F. (2005). Affect-based indexing and retrieval of films. In *Proceedings of the 13th ACM international conference on multimedia.*

Chen, H. C., & Chen, A. L. P. (2001). A music recommendation system based on music data grouping and user interests. In *Proceedings of the 10th ACM international conference on information and knowledge management.*

Gabrielsson, A., & Lindstrom, E. (2001). The influence of musical structure on emotional expression. In P. N. Juslin & J. A. Sloboda (Eds.), *Music and emotion: Theory and research.* Oxford University Press.

Giannetti, L. (2004). *Understanding movies.* Prentice Hall.

Hanjalic, A., & Xu, L. Q. (2005). Affective video content representation and modeling. *IEEE Transactions on Multimedia, 7*(1), 143–154.

Hsu, J. L., Liu, C. C., & Chen, A. L. P. (1998). Efficient repeating pattern finding in music databases. In *Proceedings of the seventh ACM international conference on information and knowledge management.*

Kalinak, K. (1992). *Settling the score: Music and the classical Hollywood film.* University of Wisconsin Press.

Kang, H. B. (2003). Affective content detection using HMMs. In *Proceedings of the 11th ACM international conference on multimedia.*

Kuo, F. F., & Shan, M. K. (2002). A personalized music filtering system based on melody style classification. In *Proceedings of IEEE international conference on data mining.*

Moncrieff, S., Dorai, C., & Venkatesh, S. (2001). Affect computing in film through sound energy dynamics. In *Proceedings of the ninth ACM international conference on multimedia.*

Ortony, A., Clore, G. L., & Collins, A. (1988). *The cognitive structure of emotions.* Cambridge University Press.

Pan, J. Y., Yang, H. J., Faloutsos, C., & Duygulu, P. (2004). Automatic multimedia cross-modal correlation discovery. In *Proceedings of ACM international conference on knowledge discovery and data mining.*

Reilly, W. S. N. (1996). Believable social and emotion agents. Doctorial dissertation. Carnegie Mellon University.

Salway, A., & Graham, M. (2003). Extracting information about emotions in films. In *Proceedings of the 11th ACM international conference on multimedia*.

Shardanand, U., & Maes, P. (1995). Social information filtering: Algorithms for automating 'word of mouth'. In *Proceedings of the conference on human factors in computing systems*.

Tao, L., & Ogihara, M. (2004). Content-based music similarity search and emotion detection. In *Proceedings of the international conference on acoustics, speech, and signal processing*.

Uitdenbogerd, A., & Zobel, J. (1999). Melodic matching techniques for large music databases. In *Proceedings of the seventh ACM international conference on multimedia*.

Wei, C. Y., Dimitrova, N., & Chang, S. F. (2004). Color-mood analysis of films based on syntactic and psychological models. In *Proceedings of IEEE international conference on multimedia and expo*.

Xu, M., Chia, L. T., & Jin, J. (2005). Affective content analysis in comedy and horror videos by audio emotional event detection. In *Proceedings of IEEE international conference on multimedia and expo*.

Yoshii, K., Goto, M., Komatani, K., Ogata, T., & Okuno, H. G. (2006). Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences. In *Proceedings of international symposium on music information retrieval.*