Tempo and beat tracking for audio signals with music genre classification

Mao-Yuan Kao and Chang-Biau Yang*

Department of Computer Science and Engineering,
National Sun Yat-sen University,
No. 70, Lienhai Rd., Kaohsiung, Taiwan
E-mail: kaomy@par.cse.nsysu.edu.tw
E-mail: cbyang@cse.nsysu.edu.tw
*Corresponding author

Shyue-Horng Shiau

Department of Computer Aided Media Design,
Chang Jung Christian University,
No. 396, Chang Jung Rd., Sec.1, Kway Jen, Tainan, Taiwan
E-mail: shiaush@mail.cjcu.edu.tw

Abstract: Most people follow the music to hum or the rhythm to tap sometimes. We may get different meanings of a music style if it is explained or felt by different people. Therefore we cannot obtain a very explicit answer if there is no music notation. Tempo and beats are very important elements in the perceptual music. Therefore, tempo estimation and beat tracking are fundamental techniques in automatic audio processing, which are crucial to multimedia applications. We first develop an artificial neural network to classify the music excerpts into the evaluation preference. And then, with the preference classification, we can obtain accurate estimation for tempo and beats, by either Ellis's method or Dixon's method. We test our method with mixed data set which contains ten music genres from the 'ballroom dancer' database. Our experimental results show that the accuracy of our method is higher than only one individual Ellis's method or Dixon's method.

Keywords: tempo; beat; audio processing; neural network; classification.

Reference to this paper should be made as follows: Kao, M-Y., Yang, C-B. and Shiau, S-H. (2009) 'Tempo and beat tracking for audio signals with music genre classification', *Int. J. Intelligent Information and Database Systems*, Vol. 3, No. 3, pp.275–290.

Biographical notes: Mao-Yuan Kao received his MS in Department of Information Engineering form I-Shou University, Kaohsiung, Taiwan in 2005 and his MS in Department of Computer Science and Engineering National Sun Yat-sen University, Kaohsiung, Taiwan in 2007. He is working in the area of audio at Brogent Corp.

Chang-Biau Yang received his BS in Electronic Engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1982 and his MS in Computer Science from National Tsing Hua University, Hsinchu, Taiwan, in 1984. Then, he received his PhD in Computer Science from National Tsing Hua University in 1988. He is currently a Professor in the Department of Computer Science and Engineering, National Sun Yat-sen University. His research interests include computer algorithms, interconnection networks and bioinformatics.

Shyue-Horng Shiau received his BS from the Department of Engineering Science at National Cheng Kung University, Tainan, Taiwan, in 1984, his MS from the Department of Applied Mathematics at National Sun Yat-sen University, Kaohsiung, Taiwan, in 1994 and his PhD from the Department of Computer Science and Engineering at National Sun Yat-sen University, in 2006. He joined the faculty of the Department of Computer Aided Media Design, Chung Jung Christian University, Tainan, Taiwan, as an Assistant Professor in 2006. His research interests include algorithms and parallel processing.

1 Introduction

Tempo is one of the basic music elements. It indicates the performance speed of the music. The *notation tempo* should be distinguished from the *perceptual tempo*. The notation tempo is a mark which is on the top of the general music staff. The perceptual tempo is the musical feeling tempo that is felt by a listener listening to the music. The listeners (even all of them are expert musicians) cannot come up with a common answer for excerpting the music annotation tempo. They will give you different answers if they are not familiar with the piece. They could tap in different metrical levels because the feelings of music melody for people may be different. Thus people may have different annotation tempos with an identical music.

A *beat* is a pulse which is a base unit for the sense of hearing. Thus a beat is a base time unit for a piece of music. It specifies every tick of the metronome and each tick is a beat. A beat can be regarded as a pulse wave or an event occurrence in the audio signal.

Tempo (McKinney and Moelants, 2004) is the speed of playing a piece of music. It is usually represented by *beats per minute* (BPM). In brief, we calculate it by counting the number of beats for playing the music per minute.

Since tempo and beat are very important elements in the perceptual music, tempo estimation and beat tracking are fundamental techniques for the automatic audio processing. In the recent research on tempo estimation, the file structure with midi or other representative form is emphasised. Recently, researcher process CD audio recordings directly, so the *WAV* or *Audio MPEG-1 Layer 3* (MP3) files become more popular. The approaches for tracking tempo and beats are usually to analyse audio signal with time frequency (Alonso et al., 2004; Peeters, 2005; Lacoste and Eck, 2007) or subband (Tzanetakis, 2005; Scheirer, 1998).

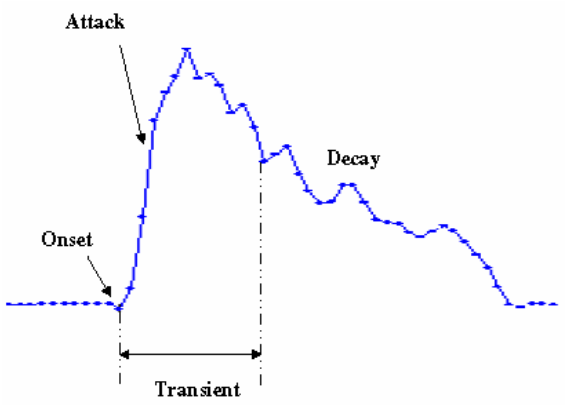
Our purpose is to detect the tempo and beat locations of the audio signal. Therefore, we first develop an artificial neural network to classify the music excerpts into the evaluation preference. And then, with the preference classification, we can obtain accurate estimation for tempo and beats, by either Ellis's method or Dixon's method.

The organisation of this paper is as follows: In Section 2, we will introduce the basic concept of the automatic audio process, Ellis's (2006) and Dixon's (2006) methods. In Section 3, we will design a neural network to classify the evaluation preference of an input music excerpt. Then, either Ellis's method or Dixon's method is invoked for tempo calculation and beat tracking. In Section 4, our experimental results will be shown and some discussions will be given. At last, in Section 5, we will give the conclusion of this paper.

2 Preliminary

An ideal sound wave signal is constituted by four parts: *onset*, *attack*, *transient* and *decay* (Bello et al., 2005), as shown in Figure 1.

Figure 1 Four elements of a signal note: onset, attack, transient and decay (see online version for colours)



The onset detection is useful for analysis technology and it can be applied to process the audio signal. It is usually used to look for transient regions on audio signal. For the realistic situation, getting an ideal audio wave is very difficult. In a real music, the signal is polyphonic, which means that many sound objects occur simultaneously in one time interval. In general, it is unable to detect the onset location directly with quantitative time-varying in the transient region. Therefore, the onset positions found by different algorithms may not be the same.

The structures in most automatic audio processing are very similar with each other. Alonso et al. (2004) presented the construction method for the automatic audio processing as follows.

Input an audio signal (waveform)

Output the tempo and the beat locations of the audio signal.

- Step 1 transform the input signal from time domain to frequency domain
- Step 2 extract the significant events from music (onset detection)
- Step 3 estimate the periodicity
- Step 4 detect the beat locations and marker.

2.1 Ellis's method for beat tracking

Ellis (2006) presented a beat tracking system with dynamic programming (Lee et al., 2005). The first stage of the system processing is to detect the onset locations of the audio signal. The original signal is down-sample to 8 kHz and it is also converted to mono. In order to keep the onset information, the system uses the half-wave rectifier to process

each frequency channel then it sums all information of the frequency channels. The system finally uses the high-pass filter to remove DC offset.

The second stage is a tempo estimation through *autocorrelation function* (ACF). In Ellis's method (Ellis, 2006), Gaussian window with a log-time axis is used to capture the important information. It has the property that the largest position is the tempo (BPM) of the audio signal. Then we can obtain the global tempo, which is important information for beat tracking processing.

The beat tracking system attempts to find out a sequence of the beat times that correspond to the large values in the onset locations of the audio signal. The best cumulative scores are found. They showed that the beat sequence consists of the ends of all possible time samples. The beat tracking system searches a range from 0.5 to 2 beat periods for each time point. And a dynamic programming approach is applied to the searching process. The best predecessor beats at the current time is obtained by choosing the largest value from the range. The value is the current onset signal value added to the best cumulative score. The time point is also stored. After the end of the audio signal, we can choose the best cumulative score and then trace back to obtain the entire sequence of beats through all the time records.

2.2 Dixon's method: beatroot

Beatroot, proposed by Dixon (2006), is a system using multiple agents' architecture. A clustering algorithm looks for the most meaningful metrical units. Based on the way, we can use the multiple agents' architecture to match the sequence of beats with the audio signal, where each agent represents a particular tempo and a sequence of beats with the audio signal.

The onset detector is to find out the peaks in the spectral flux. The spectral flux sums the magnitude changes for each frequency bin. With the onset information, the tempo estimation stage finds out a suitable beat period. The tempo estimation is first calculated by the inter-onset intervals (IOIs), where IOI is defined as an interval between any two onsets which are not necessarily consecutive. Then, a clustering algorithm is used to find out the cluster of IOIs which represent many distinct musical units. With the information, the system combines the information between the clusters on the next stage and passes the combined information to the beat tracking subsystem.

The beat tracking subsystem uses the multiple agents' architecture to find out a sequence of the beat locations which matches the varied tempo (IOI). Each agent has a tempo which is from the tempo estimation subsystem and it also has an onset time at the beginning. Each agent predicts future beats according to the given tempo and the first onset time. Each agent uses a tolerance window to find out the beat times. If an onset falls in the inner window of the predicting beat times, this onset is the real beat time. If it falls in the outer window, it is the probable beat time.

3 Our method

In our method, we first build a neural network model (Anderson, 1995; Zurada, 1995; Kumar, 2004; Abdi, 1994) to classify a given music excerpt (audio signal) into the preference of either Ellis's method (Ellis, 2006) or Dixon's method (Dixon, 2006). If the preference of an audio signal is Ellis's method, then we think Ellis's method is the better

one for estimating the tempo and beat locations of the input audio signal and thus Ellis's method is invoked. If the preference is Dixon's method, then it is done similarly.

The neural network, combining with Ellis's and Dixon's methods, can be used to detect the tempo and beat locations of an audio signal. Because our goal is to establish a neural network model, we need design the input-output pairs of the neural network. The input is the features of music genre, which can be served as the criteria of the evaluation preference classification. A binary bit will be output as the answer of the preference classification and it indicates that either Ellis's method or Dixon's method is preferred to be used for detecting the tempo and beat locations. If Ellis's method (Dixon's method) is the preferred one, then we think Ellis's method (Dixon's method) is an effective method.

Table 1 shows that the prediction accuracies of the two methods are various in different music genres. The fields of 'Ellis' and 'Dixon' indicate the accuracies of tempo prediction when Ellis's method and Dixon's method are applied, respectively. The field of 'both' means the accuracies that both Ellis's and Dixon's methods do correct prediction. Therefore, the classification of the music genre may be a good clue for us to decide which prediction method is preferred to be applied. We use the public 'ballroom dancer' music database as our training and testing data set, which can be obtained from the website (http://mtg.upf.edu/ismir2004/contest/tempoContest/node5.html).

Table 1 The accuracy of tempo and beat prediction on ten music genres with Ellis' and Dixon's methods within 1% error tolerance

Music genre	*Ellis*	*Dixon*	*Both*	*Number*
ChaChaCha	78.38%	77.48%	75.68%	111
Jive	55%	83.33%	53.33%	60
QuickStep	62.2%	8.54%	7.32%	82
Rumba American	0%	28.57%	0%	7
Rumba internation	82.35%	50.98%	49.02%	51
Rumba music	67.5%	22.5%	22.5%	40
Samba	52.33%	47.67%	27.91%	86
Tango	44.19%	72.09%	40.7%	86
Viennese waltz	35.38%	61.38%	30.77%	65
Waltz	17.27%	8.18%	2.73%	110
All data	52.29%	47.56%	34.1%	698

3.1 Feature extraction

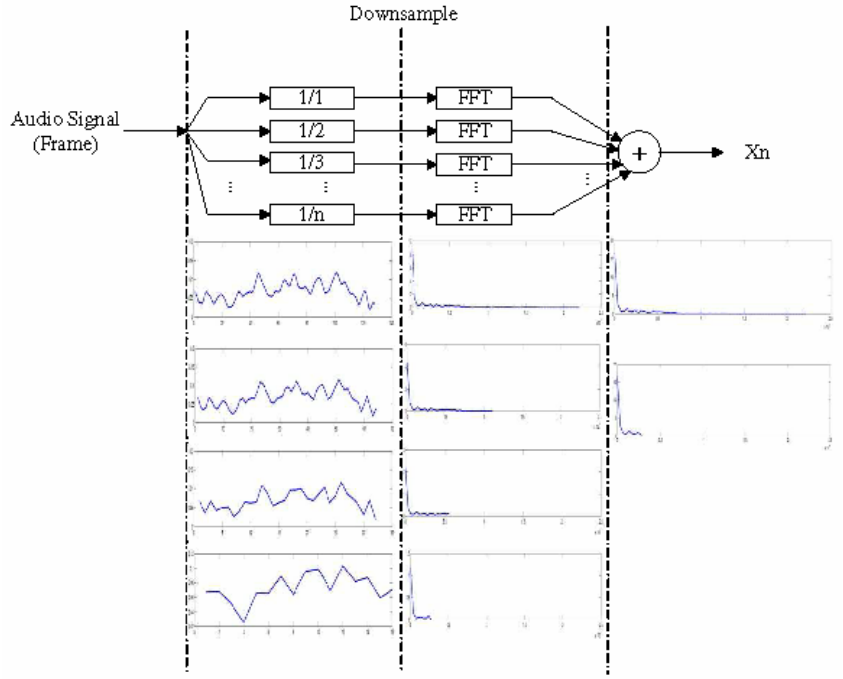
The common features extracted from the input audio signal include the following six kinds, which are the inputs of our classification neural network.

1 *Linear predictive coefficients* (LPC) (Scott, 2001): The basic principle of LPC is that the current signal value can be represented with the linear combination of the former signal values. This assumption is reasonable because most audio signals have the periodicity property. The mathematical expression of LPC is given as follows.

$$x_{n+1} = w_0 * x_n + w_1 * x_{n-1} + \cdots + w_{p-1} * x_{n-p+1} + e_{n-1}, \tag{1}$$

where e_{n+1} denotes the prediction error, which is the difference between the real sample value and predicted value. e_{n+1} is minimised by the weighted coefficients w_i in the least squares sense. LPC is called the p *linear prediction* if it predicts the current value with the p past sample values. The linear prediction is one of the important ways to represent speech signal waveform. The well-known method is Levinsion-Durbin algorithm (Levinson, 1946; Durbin, 1960; Delsarte and Genin, 1986).

Figure 2 The features obtained with discrete Fourier transform (see online version for colours)



2 *Discrete Fourier transform* (DFT) (Jang, website): We first get the audio signal on time domain and then we converts it into the frequency domain with fast Fourier transform (FT) (spectrum), as shown in Figure 2. Finally, we sum up the data obtained in various down sampling schemes. This method could enhance the data of the low frequency. Therefore, we can obtain most information of the data of the low spectrum portion. Note that the important information of most signals appears in the low spectrum portion.

3 *Harmonic product spectrum* (HPS) (Jang, website): HPS is a method of detecting the pitch in the frequency domain. It first converts the data into the frequency domain with fast FT. Then, the spectrum is down sampled continuously. Finally, HPS is the sum of all data in all downsampling steps.

4 *Cepstrum* (Jang, website): The spectrum may have some defect because the signal is easily influenced by some noises. Therefore, besides the data extracted from the frequency domain, we obtain the data of the signal calculated by the cepstrum method. The original definition of cepstrum is the *FT* of the *decibel spectrum*. To put

it plainly, the cepstrum (of signal) is calculated with the FT of log of the spectrum which is the FT of the input signal. Another definition is the *inverse Fourier transform* (IFT) of log of the decibel spectrum. The latter definition is more commonly used.

5 *Mel-scale frequency cepstral coefficients* (MFCC) (Jang, website): This parameter considers human perception sensitivity in terms of different frequency. Therefore, it is usually used on speech recognition. The process of MFCC is given as follows.

Step 1 Get the spectrum of the signal with the FT (the data of frequency domain).

Step 2 Get the log energy of each triangular band-pass filter through the magnitude frequency response multiplied by a set of N triangular band-pass filters. The relationship between the mel-frequency $\left(mel\left(f\right)\right)$ and general frequency $\left(f\right)$ is given as follows. In this paper, we set $N = 20$, which is the number of triangular band-pass filter.

$$mel\left(f\right) = 1125 * \ln\left(1 + f / 700\right) \tag{2}$$

Step 3 Take P mel-scale cepstral coefficients by applying *discrete cosine transform* (DCT) on log energy from Step 2. In this paper, we set $L = 12$, which is the number of mel-scale cepstral coefficients.

6 *Volume* (Jang, website): Volume is the strength of sound. It is also called the dynamics, intensity or energy of sound. The volume can be calculated by the amplitude of signals in each frame. Basically, there are two methods for calculating volume as follows.

1 It is the sum of absolute sample values in each frame as follows.

$$Volume = \sum_{i=1}^{n} \left| x_i \right|, \tag{3}$$

where x_i is a sample of frame and n is a frame size. This method needs only integer operations, which make this method easier than the other one.

2 It is calculated as follows.

$$Volume = 10 * log_{10}\left(\sum_{i=1}^{n} x_i^2 \right), \tag{4}$$

where x_i is a sample of frame and n is a frame size. The volume calculated by this method is based on *decibel*. It is relatively a value of intensity, which is more applicable to human perception toward the volume of sound.

3.2 Framework of our method

The overview of our method for predicting tempo and beat locations of music is given as follows.

The tempo and beat prediction method

Input An audio signal (waveform)

Output The tempo and beat locations of the audio signal.

Step 1 extract features

Step 2 preprocess (normalise and reduce the input data)

Step 3 build the neural network model

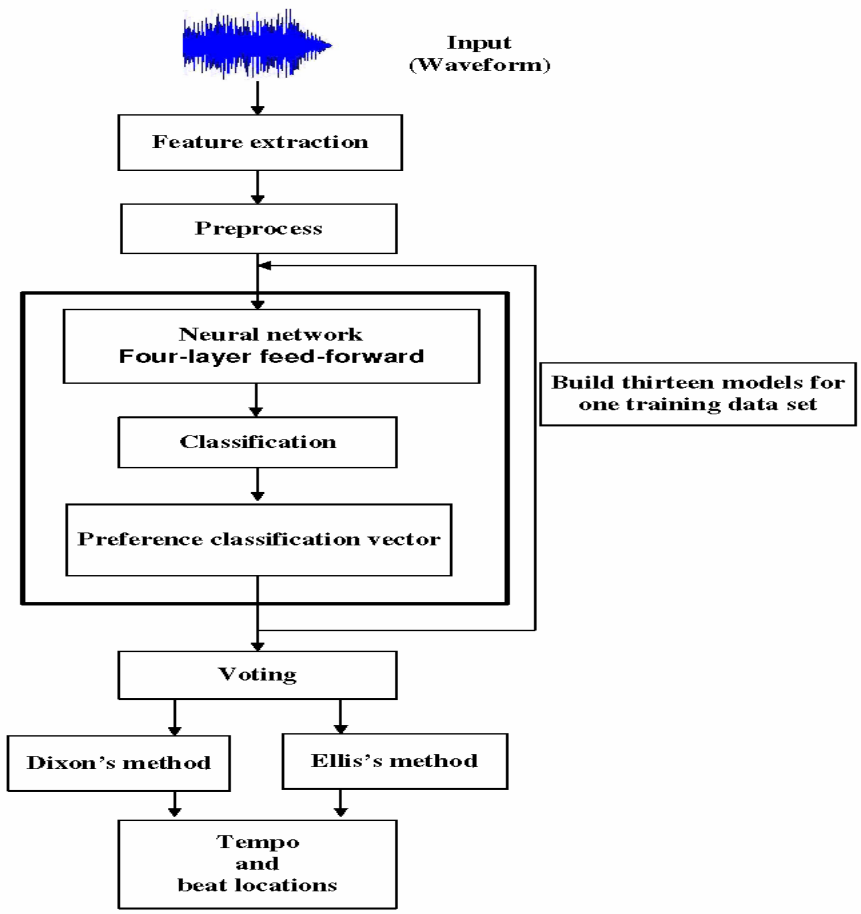
Step 4 perform classification

Step 5 vote

Step 6 predict with either Ellis's method or Dixon's method.

The flow chart of our prediction method is shown in Figure 3. The input is an audio signal and the output is the tempo and beat locations of the audio signal. The training and testing data set is the public 'ballroom dancer' music database.

Figure 3 The flow chart of our prediction method (see online version for colours)



In Step 1, the original input audio signal can not be used directly since the quantity of data is very huge. For example, the sample rate of an ordinary music (CD audio recordings) is 44,100 Hz. That is, there are 44,100 samples per second, which is too huge. For this reason, we extract the meaningful feature vectors of the audio signal to represent the audio signal. As we have already introduced, the size of each feature type in our method is shown in Table 2.

Table 2 The feature type and the number of features in each type

Feature type	*LPC*	*DFT*	*HPS*	*Cepstrum*	*MFCC*	*Volume*	*Total*
Feature number	223	359	383	359	71	12	1407

We first divide an audio signal into *N* frames in the feature extraction processing. The size of each frame is 512 samples and the overlapping size of two neighbouring frames is 128 samples. Each frame forms an operation unit. We extract 32 LPC values from each frame independently. We form the ith group of LPC values by collecting the ith elements of the 32 LPC values of all frames together. Thus, 32 groups are obtained. Then, in each group, we calculate its mean, sum, maximum, minimum and median values. Now, each group is represented by these five values. In order to extract the relationship among all groups, we first calculate the *N* differences between the corresponding pairs of two groups i and $i+1$, where $1 \le i \le 31$. Then the maximum of the *N* differences between groups i and $i+1$ is calculated as the difference of these two groups. Thus, we get 31 difference values between every two neighbouring groups. In addition, we extract 32 LPC values from the whole audio signal. Summing the above amounts, we have 32 * 5 + 31 + 32 = 223 values of LPC. Thus the LPC values contain important information for time domain.

To reduce the noise in time domain, we extract important information from frequency domain which contains low and high frequency data (DFT, HPS and cepstrum). We also extract 60 values containing 50 elements of low frequency and ten elements of high frequency in the DFT step, 64 values containing 32 elements of low frequency and 32 elements of high frequency in the HPS step, 60 values containing ten elements of low frequency and 50 elements of high frequency in the cepstrum step and 12 MFCC values from each frame. We also group the above values and then use the five values (mean, sum, maximum, minimum and median) to represent each group. The maximum value among the *N* difference values between every two adjacent groups are also calculated. Thus we have the number of the discrete FT amplitude values is 60 * 5 + 59 = 359, the number of the HPS coefficients is 64 * 5 + 63 = 383, the number of the cepstrum coefficients is 60 * 5 + 59 = 359 and the number of the MFCC is 12 * 5 + 11 = 71. We also extract two volume values for each frame. Since we use two methods to calculate the maximum value, the number of the volume coefficients is 2 * 5 + 2 = 12. Summing the above amounts, 223 + 359 + 383 + 359 + 71 + 12, we have 1407 which is the total number of features in the six types extracted in Step 1. The MFCC and volume values can be regarded as the characteristic features for the sound.

The 1407 features in the six types extracted in Step 1, is still huge for a neural network. Therefore, in Step 2, we have to reduce the dimension of the feature vector, where a dimension represents a feature type. There are 698 music excerpts in the 'ballroom dancer' music database. We divide the data into three disjoint sets, one for

training another for validation and the other for testing. Then, we normalise the training and validation data sets by regulating their average values and the standard deviations. The normalisation procedure is given as follows.

The $meanp$ is the average value of the input vector:

$$meanp = \overline{p} = \frac{1}{n}\sum_{i=1}^{n} p_i. \tag{5}$$

The $stdp$ is the standard deviation of the input vector:

$$stdp = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(p_i - \overline{p}\right)^2}. \tag{6}$$

Then, the input vector can be normalised as follows:

$$p' = \left(p - meanp\right)/stdp, \tag{7}$$

where p is the input vector and p' is the normalised result of p. The $meant$ is the average value of the target vector:

$$meant = \overline{t} = \frac{1}{n}\sum_{i=1}^{n} t_i. \tag{8}$$

The $stdt$ is the standard deviation of the target vector:

$$stdt = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(t_i - \overline{t}\right)^2}.$$

Then, the target vector can be normalised as follows:

$$t' = \left(t - meant\right)/stdt, \tag{9}$$

where t is the target vector and t' is the normalised result of t.

We can also normalise the testing data set with $meanp$, $stdp$, $meant$ and $stdt$ similarly. Then, we apply principal components analysis (PCA) (Smith, 2002) to the training and validation data sets for reducing their dimensions to lower dimensions as follows.

Step 1 find a set of orthogonal principal components from the input vector (feature vector)

Step 2 sort the principal components according to the variation degree

Step 3 remove the principal components which have the least contributions.

The principal components can be calculated as follows. First, the covariance matrix of input vector *p* is calculated by

$$S = \frac{1}{n-1}\sum_{i=1}^{n}\left(p_i' - \overline{p}'\right)\left(p_i' - \overline{p}'\right)^T. \tag{10}$$

Then, the eigenvectors U and eigenvalues V of the covariance matrix can be obtained by

$$S = UVU^T \tag{11}$$

Finally, we get the principal components as follows.

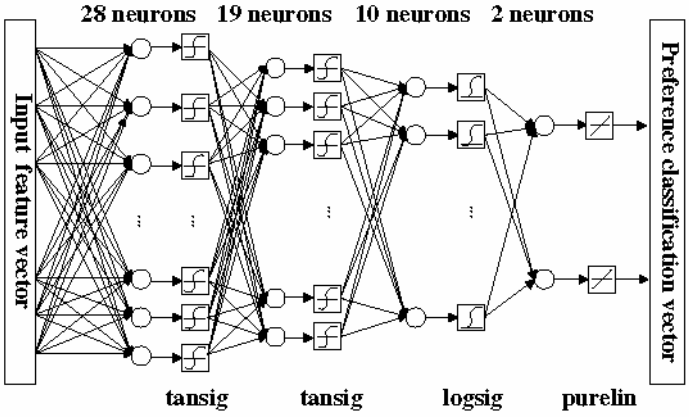
$$c = U^T p', \tag{12}$$

where p' is the normalised result of the input vector p. We use the principal components whose accumulated contribution exceeds 99.9% of the variation in the training data set to build the neural network model. Table 3 shows the dimension in each feature type remained after PCA. Matrix U^T is also used to get the principle components of the testing data set.

Table 3 The dimension of each feature type after PCA with threshold 99.9% (average)

Feature type	*LPC*	*DFT*	*HPS*	*Cepstrum*	*MFCC*	*Volume*	*All*
Feature number (PCA)	31	67	65	99	31	10	80

Figure 4 A four-layer neural network



In Step 3, we build a four-layer *feed-forward back-propagation* neural network (Kumar, 2004), as shown in Figure 4. The four-layer neural network is more stable than the three-layer. If the number of layers increases from four to five, the execution time will be raised, but the gain in the accuracy improvement will be relatively small. The numbers of neurons in the input, second, third and output layers are 28, 19, 10 and 2, respectively. Before the endings of the input and second layers, *tangent sigmoid transfer function* (Kumar, 2004) is used to convert the calculated element to fall within the range from –1 to 1 as follows:

$$f_{tan}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{13}$$

The third layer applies the *log sigmoid transfer function* (Kumar, 2004) to convert the result to fall within the range from 0 to 1 as follows:

$$f_{log}(x) = \frac{1}{1+e^{-x}}. \tag{14}$$

No transfer is done in the output layer.

The *training function* we use is the *Levenberg-Marquardt back-propagation algorithm* (Levenberg, 1946; Marquardt, 1963; Roweis, website; Kumar, 2004) which is an iterative procedure. The algorithm is a second conjugate gradient method that is a quick optimisation algorithm. The expression used in the algorithm is given as follows:

$$X_{k+1} = X_k - \left[J^T J + \mu I\right]^{-1} J^T e, \tag{15}$$

where J is a Jacobian matrix which includes the first-order partial derivatives of the weight and bias values with respect to the network error vector e.

The *performance function* we use is the mean squared error with regularisation performance function (Zurada, 1995; Kumar, 2004; Abdi, 1994) which estimates the performance of the network with the mean squared error and the mean squared weight and bias. The calculation of mean squared error is given as follows:

$$\frac{1}{n}\sum_{i=1}^{n}(e_i)^2. \tag{16}$$

The calculation of mean squared weight and bias is given as follows:

$$\frac{1}{n}\sum_{j=1}^{n}(w_j)^2. \tag{17}$$

Then the mean squared error with regularisation is calculated as follows:

$$\gamma\frac{1}{n}\sum_{i=1}^{n}(e_i)^2 + (1-\gamma)\frac{1}{n}\sum_{j=1}^{n}(w_j)^2, \tag{18}$$

where γ is used to weight the two terms.

The *adaptive learning function* we use is the *gradient descent with momentum* weight *and bias learning function* (Zurada, 1995; Kumar, 2004; Abdi, 1994) which calculates the weight change dw according to gradient descent with momentum:

$$dw = mc * dw_{prev} + (1-mc) * lr * gw, \tag{19}$$

where lr is learning ratio, mc is momentum constant, gw is the weight from gradient descent and dw_{prev} is the previous weight change.

In Step 4, we use two binary bits to represent the classification result predicted by the neural network. '01' means that Ellis's method is the better one for predicting tempo and beat locations for the input music signal and '10' means that Dixon's method is the better one.

Since each neural network model is not very reliable, to increase the classification accuracy, we build 13 neural network models. Then the classification result is determined by the vote of the 13 models with the majority rule. Thus, Step 5 performs the voting procedure.

According to the voting result, Step 6 invokes either Ellis's method or Dixon's method to predict the tempo and beat locations of the input audio signal. Finally, our method outputs the estimated tempo and the predicted beat location by the result obtained from either Ellis's method or Dixon's method.

4 Experimental results

Our classification method with the neural network is implemented by Matlab 7.1.0.246 (R14) service pack 3. Our simulation environment is a PC with AMD Sempron(tm) 2600+ as processor and 512 MB DDR RAM.

We evaluate our method with a public 'ballroom dancer' database. Suppose the tempo of a music excerpt is estimated by a method. If the absolute difference between the estimated tempo and the real tempo is not greater than P times the real tempo, we say that the estimation of the music excerpt is correct. Usually the error tolerance is $P = 0.01$. There are ten music genres (types) in the 'ballroom dancer' database, which contains 698 music excerpts (music segments). Here, the training, validation and testing data sets are randomly chosen from the 'ballroom dancer' database with sizes about $\frac{1}{2}, \frac{1}{4}$ and $\frac{1}{4}$ of each music genre, respectively. For the experiment purpose, the genre of each excerpt in the training set is assumed to be known. But, the genre of each excerpt in the validation or testing set is assumed to be unknown. When we build a neural network, we try to give three kinds of different parameter settings as follows.

All-parameter: All parameters included in library function for the neural network in Matlab are set as follows:

$max\_fail = 5,$ the maximum occurrences of the validation failures.

$mem\_reduc = 1,$ the factor for memory and speed trade-off.

$mu = 1,$ the initial μ value.

$mu\_dec = 0.8,$ the decreasing factor of μ.

$mu\_inc = 1.5,$ the increasing factor of μ.

$mu\_max = 1e^{10},$ the maximum value of μ.

Validation convergence: In addition to the above parameters, the training program may terminate early due to the convergence of the validation data set.

Some-parameter: The settings in all-parameters and the validation data set are removed. Instead, default settings are used.

We build 13 neural network models for each of the above three parameter settings. Then, we use a voting mechanism to get the final result. In Table 4, we use 'All', 'Val' and 'Some' to represent 'all-parameter', 'validation convergence' and 'some-parameter', respectively. In each experiment, we form the training, validation and testing data sets by randomly selected from the database. The evaluation preference is decided by the vote of 13 neural network models, each with the three types of parameter settings. In this paper, we perform the above experiment 14 times. The mean and standard deviation of the 14

experimental results is shown in Table 4. The table shows that the all-parameter setting gets more accurate and more stable solutions. The accuracy of our method is better than the other two methods. In fact, our method gets better solutions in all of the 14 experiments.

Table 4 The prediction accuracies based on the preference classification of ten music genres

	Elli's method	*Dixon's method*	*Our method (All) (%)*	*Our method (Val) (%)*	*Our method (Some) (%)*
Training data set (mean)	53.043	47.571	60.473	57.873	60.505
Testing data set (mean)	51.275	46.378	54.625	53.522	53.829
Validation data set (mean)	52.886	49.038	55.326	55.083	55.302
Training data set (standard deviation)	1.825	1.059	1.344	1.745	1.361
Testing data set (standard deviation)	2.370	2.197	2.624	3.101	3.218
Validation data set (standard deviation)	2.466	2.804	3.045	2.846	2.166

In addition, we perform another experiment in which the original input data set is changed. We change the classification from the ten music data sets (genres) of the 'ballroom dancer' database to four data sets. The four data sets represent four types which are '01', '10', '00' and '11'respectively. '01' means that Ellis's method is the better one for predicting tempo and beat locations for the input music signal and '10' means that Dixon's method is the better one. '00' means that neither of Ellis's or Dixon's methods can get correct prediction. '11' means that both Ellis's and Dixon's methods can get correct prediction.

We choose the training and testing data sets from the four data sets for insuring that the size of the training data set is enough. The training data set is constructed from the '01' and '10' data sets and the testing data set is randomly chosen from the four data sets with sizes from $\frac{1}{4}$ to $\frac{1}{2}$ of the each data set. The training way and the estimated method are the same as the above description. We also use a neural network to classify the music excerpts into the evaluation preference. And then, with the preference classification, we can obtain accurate estimation for tempo and beats, by either Ellis's method or Dixon's method. We set the evaluation preference to Ellis's method if the music excerpt is not classified by neural network. We perform the experiment 14 times. The mean and standard deviation of the 14 experimental results is shown in Table 5. We also get better solutions in all of the 14 experiments by this way.

Table 5 The prediction accuracies based on the preference classification of four types

	Elli's method (%)	*Dixon's method (%)*	*Our method (All) (%)*
Testing data set (mean)	54.179	47.577	56.336
Testing data set (standard deviation)	4.727	5.069	4.345

5 Conclusions

We first build a neural network model to classify a given music excerpt (audio signal) into the preference of either Ellis's (2006) method or Dixon's (2006) method. As the experimental results described in Section 4, the accuracy of our method is better than only one individual Ellis's method or Dixon's method. It is not easy to extremely increase the accuracy of detecting the tempo and beat locations of the audio signal. Moreover, it is difficult to detect the tempo and beat locations of the audio signal for some music genres such as the classical music, jazz music, etc. Detecting the tempo and beat locations by using only one method is certainly unable to obtain a good result. Thus our method is to build a system combining the advantages of the two methods to obtain a good result for each music genre.

Acknowledgements

This research work was partially supported by the National Science Council of Taiwan under contract NSC 95-2745-H-309-003-HPU.

References

Abdi, H. (1994) 'A neural network primer', *Journal of Biological Systems*, Vol. 2, No. 3, pp.247–283.

Alonso, M.A., David, B. and Richard, G. (2004) 'Tempo and beat estimation of musical signals', *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, Barcelona, Spain.

Anderson, J.A. (1995) *An Introduction to Neural Networks*, MIT Press.

Bello, J.P., Daudet, L., Abdallah, S., Duxbury, C., Davies, M., Sandler, M.B. and Member, S. (2005) 'A tutorial on onset detection in music signals', *IEEE Transactions on Speech and Audio Processing*, Vol. 13, No. 5, pp.1035–1047.

Delsarte, P. and Genin, Y.V. (1986) 'Establishing homologies in protein sequences', *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, No. 3.

Dixon, S. (2006) 'MIREX 2006 audio beat tracking evaluation: BeatRoot', *Proceedings of the 2th Music Information Retrieval Evaluation eXchange (MIREX 2006)*.

Durbin, J. (1960) 'The fitting of time series models', *Revue de l'Institut International de Statistique 28*, Vol. 28, pp.233–243.

Ellis, D.P.W. (2006) 'Beat tracking with dynamic programming', *Proceedings of the 2th Music Information Retrieval Evaluation eXchange (MIREX 2006)*.

Jang, J.S.R. (website) *Audio Signal Processing and Recognition*, in Chinese, available at http://neural.cs.nthu.edu.tw/jang.

Kumar, S. (2004) *Neural Networks: A Classroom Approach*, McGraw-Hill.

Lacoste, A. and Eck, D. (2007) 'A supervised classification algorithm for note onset detection', *EURASIP Journal on Advances in Signal Processing*, Vol. 2007, No. 43745, p.13.

Lee, R.C.T., Tseng, S.S., Chang, R.C and Tsai, Y.T. (2005) *Introduction to the Design and Analysis of Algorithms*, McGraw-Hill.

Levinson, N. (1946) 'The wiener root-mean-square error criterion in klter design and prediction', *Journal of Mathematics and Physics*, Vol. 25, pp.261–278.

Marquardt, D. (1963) 'An algorithm for the least-squares estimation of nonlinear parameters', *SIAM Journal on Applied Mathematics*, Vol. 11, No. 2.

McKinney, M.F. and Moelants, D. (2004) 'Deviations from the resonance theory of tempo induction', *Proceedings of the Conference on Interdisciplinary Musicology (CIM04)*, Graz, Austria.

Peeters, G. (2005) 'Time variable tempo detection and beat marking', *Proceedings of International Computer Music Conference (ICMC2005)*, Barcelona, Spain.

Roweis, S. (website) *Probably Useless Notes: Levenberg-Marquardt Optimization*, available at http://www.cs.toronto.edu/»roweis/notes.html.

Scheirer, E.D. (1998) 'Tempo and beat analysis of acoustic musical signals', *The Journal of the Acoustical Society of America*, Vol. 103, pp.588–601.

Scott, P. (2001) *Music Classification Using Neural Networks*, Stanford University.

Smith, L.I. (2002) *A Tutorial on Principal Components Analysis*, available at http://csnet.otago.ac.nz/cosc453/student tutorials/principal components.pdf.

Tzanetakis, G. (2005) 'Tempo extraction using beat histograms', *Proceedings of the 1st Music Information Retrieval Evaluation eXchange (MIREX 2005)*.

Zurada, J. (1995) *Introduction to Artificial Neural Systems*, West Publishing Co. St. Paul, MN, USA.