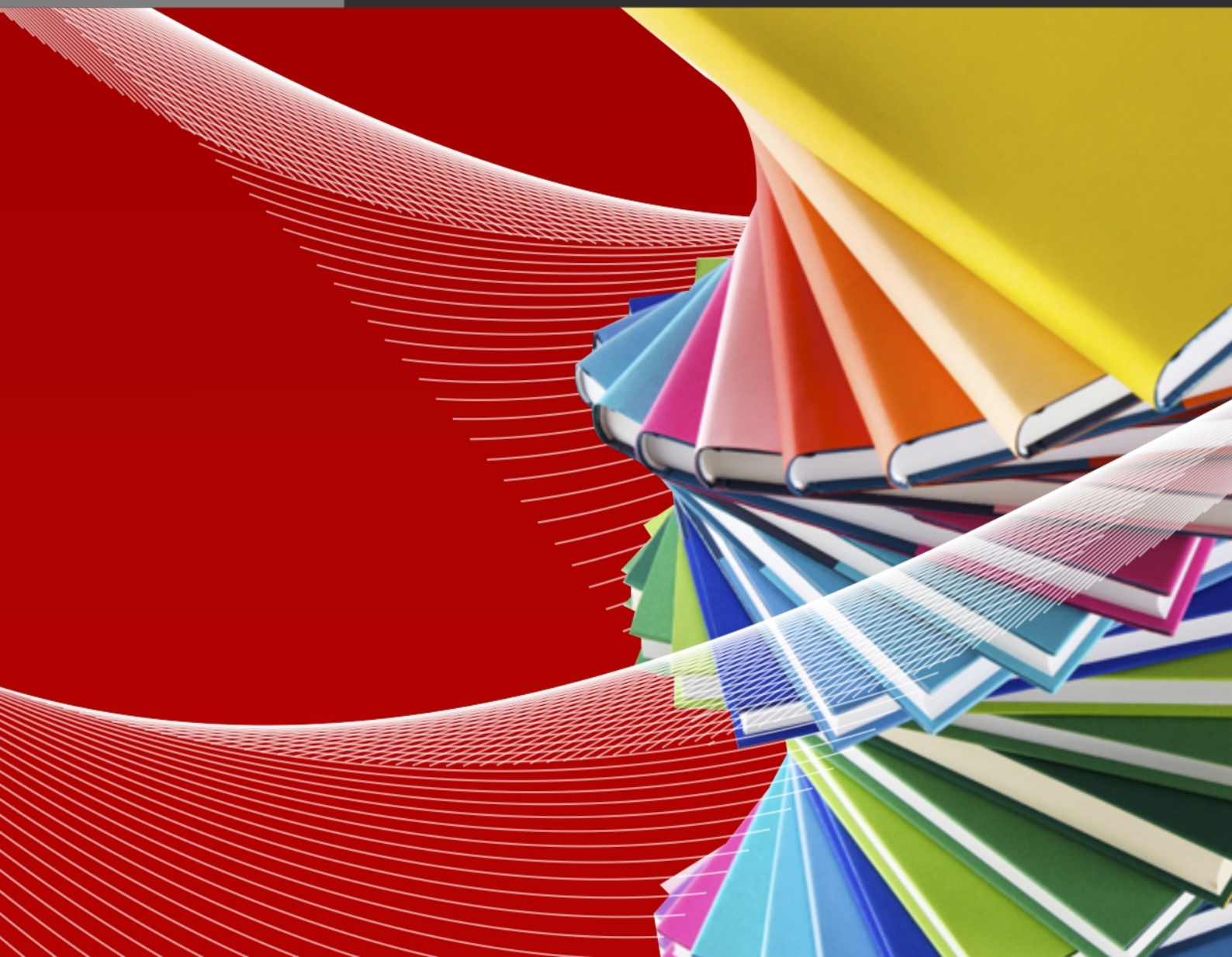
澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

Outstanding Academic Papers by Students
學生優秀作品



User Customization for Music Emotion Classification using Online Sequential Extreme Learning Machine

by

CHI-MAN WONG

(DB128151)

Supervisor: CHI-MAN VONG

Final Project Report submitted in partial fulfillment
of the requirements of the Degree of
Bachelor of Science in Computer Science

15 May 2015

DECLARATION

I sincerely declare that:

1. I and my teammates are the sole authors of this report,
2. All the information contained in this report is certain and correct to the best of my knowledge,
3. I declare that the thesis here submitted is original except for the source materials explicitly acknowledged and that this thesis or parts of this thesis have not been previously submitted for the same degree or for a different degree, and
4. I also acknowledge that I am aware of the Rules on Handling Student Academic Dishonesty and the Regulations of the Student Discipline of the University of Macau.

Signature : _____

Name : WONG CHI MAN

Student No. : DB128151

Date : 21 May 2015

UNIVERSIDADE DE MACAU
仁義禮知信
澳門大學

Abstract

USER CUSTOMIZATION FOR MUSIC EMOTION CLASSIFICATION USING ONLINE SEQUENTIAL EXTREME LEARNING MACHINE

By Chi-Man Wong

Thesis Supervisor: Dr. Chi-Man Vong

Department of Computer and Information Science

Machine learning techniques have been widely applied to handle complicated and advanced classification problem including classification of music emotion. In this work, traditional machine learning algorithms such as k-nearest neighbor, and state-of-the-art neural network methods such as support vector machine, and extreme learning machine are applied and compared with different combinations of feature sets extracted from a benchmark music set using MIRtoolbox for music emotion classification. The best classifier with the best feature sets combination is chosen for user customization. Because music emotion perception is subjective and can vary individual to individual, the model may not fit all users. To overcome this problem, we propose using online sequential extreme learning machine to update the model based on user's preference because it is fast and accurate with good generalization ability. The result showed that with respect to user's preference, the model can be updated immediately and remained a similar accuracy. Further more, it is also important to learn a good feature representation for music emotion classification task, so that we also investigate on the deep network such as Multi layer extreme learning machines (MLELM). MLELM inherits the fastness property of ELM while it can learn higher level of feature representation for classification. However, there are three problems for MLELM: 1) To determine the number of hidden neurons for each layer; 2) When the number of hidden neurons of i^{th} hidden layer is different than the number of hidden neurons of $(i+1)^{th}$ hidden layer, the output from i^{th} hidden needed to be scaled in order to fit in the $(i+1)^{th}$ hidden layer; 3) It assumes that the feature space for each hidden layer is the same. To solve these problems, we propose using a new kernel-based MLELM, to analytically determine the hidden neurons number for each layer. The result showed that the proposed kernel MLELM obtained a better performance than MLELM, SVM, and ELM.

TABLE OF CONTENTS

LIST OF FIGURES

Number *page*

LIST OF TABLES

Number *Page*

UNIVERSIDADE DE MACAU
仁義禮知信
澳門大學

LIST OF ABBREVIATIONS

MIR	Music information retrieval
FOAF	Friend of a Friend
SLFNs	Single hidden layer feedforward networks
K-NN	K-nearest neighbors
SVM	Support vector machines
ELM	Extreme learning machines
OSELM	Online sequential extreme learning machines
MIREX	Music Information Retrieval Evaluation eXchange
RMS	Root mean square
FTT	Fast Fourier transform
MFCCs	Mel-frequency cepstrum coefficients
RBF	Radial basis function
HCDF	Harmonic Change Detection Function
MLELM	Multi layer extreme learning machines
K-MLELM	Kernel-based multi layer extreme learning machines

UNIVERSIDADE DE MACAU
仁義禮知信
澳門大學

ACKNOWLEDGEMENTS

The author wishes to thank Prof. Chi-Man for giving many useful advices on this thesis. Without his help, this research cannot be completed successfully

UNIVERSIDADE DE MACAU
仁義禮知信
澳門大學

CHAPTER 1: INTRODUCTION

With the massive growth of music on the Internet, however, this becomes a problem for music listener to search for their favorite music piece in the near future. Music recommendation system becomes a bridge between music listener and music piece, helping music listener to find the music pieces that they are likely to enjoy [14, 15]. In fact, there are more and more transactions of music piece on the Internet such as iTunes music store, so that music recommendation is also important in e-World. Music recommendation is a part in the field of music information retrieval (MIR) that automated music recognition, the design and extraction of musically relevant audio features, music information handling and retrieval and classification are addressed [16].

A lot of research has been studied on music emotion recommendation [1], which is a procedure of determining which music fits to listener's desire of music emotion. Recent recommendation systems are classified into three categories: collaborative filtering [2, 17], content-based filtering [3] and hybrid approach [4]. Collaboratively filtering approaches recommend music by using listener's historical behaviors such as ratings. For example, if music listener A and B rate n music pieces similarly or have similar behaviors (such as buying and listening), they will rate on other music pieces similarly. However, this approach requires a lot of ratings data that may not be available and often suggest a famous music piece that is well known to the user listener, and thus ineffective or meaningless. Typical large online shopping companies, eBay and Amazon are using collaboratively filtering approach. Content-based filtering approaches recommend music based on the music listener's preference such as the meta-data (such as genre, styles, artists) and acoustic features (such as pitch, timbre, and rhythm) [9]. This recommendation approach can suggest a variety of music pieces, however, music listeners should explicitly list their preferences. Pandora and Friend of a Friend (FOAF) are typical music recommendation system using context-based filtering approach.

In context-based filtering approach, machine-learning methods are usually applied into music recommendation system. Acoustic features of a music piece can be extracted for classification such as root mean square (RMS) [18], Fast Fourier Transform (FTT) [19], Mel-frequency cepstrum coefficients (MFCCs) [20], auditory modeling [21]. RMS is the simplest features based on audio waveform itself. Many features can be derived from the FFT

such as brightness, spectral centroid, roughness, and sensory dissonance, etc. MFCCs describe timbre from audio frames such as pitch and intensity. However, not all acoustic features are important in music emotion classification, so that it is also important to explore the relationship between acoustic features and music emotion [23].

In music emotion classification, various machine-learning techniques have been applied such as Gaussian processes [25], k-nearest neighbors (k-NN) [5], and support vector machines (SVM) [6], but so far SVM is the dominant model for MIR task. SVM techniques maps the extracted audio features into high dimensional feature space to perform a non-linear classification. However, it requires iterative operation to find the optimal solution so that it is often expensive. In recent year, extreme learning machine (ELM) have been proposed to handle complex multi-classification problem with very fast training speed and with even better accuracy [7], which becomes an obvious appeal for multi-media processing. However, user's emotion perception is subjective and can vary individual to individual so that after the model is trained by machine learning methods, it may not fit to all users. For example, the model tagged a music with "happy" may not match user's preference, thus a fast and accurate algorithm for updating model is desired. To handle this problem, a fast and accurate online sequential learning algorithm with good generalization ability called online sequential extreme learning machine (OSELM) [8] is proposed, which can learn data one by one or chunk by chunk with fixed or varying chunk size.

Furthermore, deep networks have been proposed to learn a high-level feature representation for classification such as multi layer extreme learning machines (MLELM) [29]. MLELM is an ELM-based deep networks which inherits the fastness of ELM that once the parameters of hidden layer are randomly generated and then are analytically determined. Unlike the traditional deep network, the parameters of hidden layers for MLELM need not to be fine-tuning. However, there are three problems: 1) To determine the number of hidden neurons for each layer. For example, searching for an optimal number of hidden neurons for three layer MLELM is exhaustive; 2) When the number of hidden neurons of i^{th} hidden layer is different than the number of hidden neurons of $(i+1)^{th}$ hidden layer, the output from i^{th} hidden needed to be scaled in order to fit the $(i+1)^{th}$ hidden layer; 3) When classifying a new data, it assumes that the feature space for each hidden layer is the same and then by

multiplying the input data with all output weight of hidden layers. In this work, we propose using kernel-based multi layer extreme learning machines (kernel MLELM) to solve the above problems. Kernel MLELM can analytically determine the number of hidden neurons and unlike the classification procedure by MLELM, Kernel MLELM follows the procedure of Markov hidden model. When new data appears, the data will be mapping to a higher feature space by kernel (as state) and multiplying the respective output weight (as feature learning) for every hidden layer.

In this work, to explore the relationship between acoustic features and music emotion, we compare three machine learning algorithms: k-NN, SVM, and ELM, with four different audio features combination (that are dynamics, spectrum, rhythm, and harmony); then the best classifier with the best features sets combination is chosen and evaluate the update for model by using OSELM. The training and testing dataset are using MIREX Mood Classification dataset in which a music is tagged with an emotion. Finally, we will extend the experiment by testing the classifier MLELM and kernel MLELM.

This paper is organized as follows. Section two gives a review of *k*-NN, SVM, ELM, MLELM, and proposed kernel MLELM, and OSELM. In section 3, the experiment setting in this work as the employed benchmark dataset, music feature description, music feature extraction, different methods for parameters setup, and discussion is described. The results are given in Section 4. Finally, section 5 concludes the paper

CHAPTER 2. EMPLOYED ALGORITHMS

2.1 CLASSIFICATION

2.1.1 *k*-NEAREST NEIGHBORS

k-NN is a non-parametric method used for classification and regression [28], which is among the simplest of all machine-learning algorithms. Given a dataset $\mathbf{D} = \{(\mathbf{x}_i, y_i)|\mathbf{x}_i \in R^d, y_i \in R,\ i = 1, \dots, N\}$, finding k most training samples closest to the data point is equivalent to finding k least Euclidean distance of the training samples with data point. The Euclidean distance is defined as

$$\text{Euclidean } d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} \sqrt{(\mathbf{x}_i - y_i)^2} \tag{1}$$

2.1.2 SUPPORT VECTOR MACHINES

SVM was proposed by Vapnik [27], which is a supervised machine learning method for classification and regression over linear and nonlinear separable data. For linear case, let the N training sample $\mathbf{D} = \{(\mathbf{x}_i, y_i)|\mathbf{x}_i \in R^d, y_i \in \{+1, -1\}, i = 1, \dots, N\}$, in which $\mathbf{x}_i$ is a vector with two attributes and y_i is the target output. The hyperplane that separates the two classes can be defined as

$$\mathbf{w}\mathbf{x} + \mathrm{b} = 0 \tag{2}$$

Where $\mathbf{w}$ is the adjustable normal vector to the hyperplane with respect to the input $\mathbf{x}$, and b is the bias. The margin is defined as the distance between the hyperplane and the point closest to the hyperplane. Then, finding the maximum of the margin leads to optimal solution. The distance between separating hyperplane and the closest point is $\frac{1}{\|\mathbf{w}\|}$, where $\|\mathbf{w}\|$ is the Euclidean norm. Then, the margin is given by $\frac{2}{\|\mathbf{w}\|}$ and thus to maximize the margin is equivalent to minimize $\|\mathbf{w}\|$. By the Karush-kuhh-Tucker conditions, the minimizer becomes

$$L(\mathbf{w}, \mathrm{b}, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \alpha_i[y_i(\mathbf{w}\mathbf{x} + \mathrm{b}) - 1] \quad (3)$$

Where $\boldsymbol{\alpha_i}$ is the Lagrange multiplier. Then, we can obtain the decision function

$$f(\mathbf{x}^T) = sgn(\sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \mathbf{x}^T + \mathrm{b}) \in \{+1, -1\}(\quad (4)$$

This function can be used to two-class classification that data is linearly separate. For most cases that are non-linearly separate, SVM will map those nonlinear data into a high dimensional feature space, i.e. $\mathrm{x} \rightarrow \varphi(\mathrm{x})$. Then the decision can be written as

$$f(\mathbf{x}) = \sum_{i,j} \alpha_i y_i \varphi(\mathbf{x}_i)\varphi(\mathbf{x}_j) \quad (5)$$

However, the dimensionality is unknown so that kernel methods are applied $K(\mathbf{u}, \mathbf{v}) = \varphi(\mathbf{u})\varphi(\mathbf{v})$. Finally the decision function for non-linear classification becomes

$$f(\mathbf{x}) = sgn(\sum_{i=1}^{N} \alpha_i y_i \mathbf{k}(\mathbf{x}_i, \mathbf{x}) + \mathrm{b}) \in \{+1, -1\} \quad (6)$$

2.1.3 EXTREME LEARNING MACHINES

Huang et al. [26] proposed a single hidden layer feedforward networks (SLFNs) with random hidden nodes. Let $D = (\mathbf{x}_i, \mathrm{t}_i), i = 1 \text{ to } N$ be a set of training sample given where the input sample $\mathbf{x}_i = (\mathrm{x}_{i1}, \ldots, \mathrm{x}_{iN}) \in R^N$ and its respective target output value $y_i \in R$. The output function of ELM is represented by

$$f(x) = sign(\sum_{k=0}^{L} \beta_k \mathrm{h}_k(\boldsymbol{\alpha}_i, b_i, \mathbf{x}_i)) \quad (7)$$

Where β is the vector of the output weight of L nodes to an output node, and $\mathrm{h}(\mathbf{x})$ is the hidden feature mapping with respect to input x and the hidden node parameters $(\boldsymbol{\alpha}_i, b_i)$ can be randomly generated.

The output function of (1) can be written as

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \quad (8)$$

Where

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathrm{x}_1) \\ \vdots \\ \mathbf{h}(\mathrm{x}_N) \end{bmatrix} = \begin{bmatrix} \mathrm{h}_1(\mathrm{x}_1) & \cdots & \mathrm{h}_L(\mathrm{x}_1) \\ \vdots & \ddots & \vdots \\ \mathrm{h}_1(\mathrm{x}_N) & \cdots & \mathrm{h}_L(\mathrm{x}_N) \end{bmatrix} \tag{9}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \vdots \\ \boldsymbol{\beta}_n^T \end{bmatrix} \text{ and } \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_n^T \end{bmatrix} \tag{10}$$

H is called the hidden layer output matrix of the neural network, and ith column of **H** is the output of the ith hidden node with respect to input $\mathbf{x}_1, \ldots, \mathbf{x}_i$.

After the hidden node parameters are generated, training SLFNs is equivalent to finding a least square solution $\widehat{\boldsymbol{\beta}}$ of linear system:

$$\left\|\mathbf{H}\widehat{\boldsymbol{\beta}} - \mathbf{T}\right\| = \min\|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\| \tag{11}$$

The minimal norm least square solution is

$$\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T},\ \mathbf{H}^{\dagger} = (\mathbf{H}^{\mathbf{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathbf{T}} \tag{12}$$

where $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse of matrix **H** [10].

2.1.4 MULTI LAYER EXTREME LEARNING MACHINES

L.L.C Kasun et al. proposed multi layer extreme learning machines extreme learning machine composed by two different components: ELM-based auto encoder is to learn higher feature representation and the classification for the learned features are by ELM [29]. When the number of hidden neurons of i^{th} hidden layer is equal to the number of hidden neurons of $(i+1)^{\text{th}}$ hidden layer, the feature mapping function G=(**ax** + b) is chosen to be linear otherwise, g is chosen as nonlinear piecewise, where **a** is input weight and b is bias. The input weight **a** and bias b are chosen to be orthogonal and are randomly generated. The output weight is calculated by minimal least square solution. For feature learning, the input data is sat to be the target output, while for each hidden layer, the output weight would be $\mathbf{w}^{\boldsymbol{i+1}} = \mathbf{w}^{\boldsymbol{i}}\mathbf{H}^{\boldsymbol{i}} = (\mathrm{G}((\mathbf{w}^{\boldsymbol{i}})\mathbf{H}^{\boldsymbol{i-1}}))^{\dagger}\mathbf{x}\mathrm{G}((\mathbf{w}^{\boldsymbol{i}})\mathbf{H}^{\boldsymbol{i-1}})$, where $\mathbf{w}^{\boldsymbol{i+1}}$ is the output weight of $(i+1)^{\text{th}}$ hidden layer and $\mathbf{H}^{\boldsymbol{i}}$ is the output of mapping function G for *i* hidden layer.

2.1.5 KERNEL BASED MULTI LAYER EXTREME LEARNING MACHINES

In this work, instead of using different number of hidden neurons for each hidden layer, the proposed kernel-based multi layer extreme learning machines (kernel MLELM) use kernel trick to determine the hidden neurons for each layer. Let $D = (\mathbf{x}_t, t_i), i = 1 \text{ to } N$ be a set of training sample given where the input sample $\mathbf{x}_i = (x_{i1}, \ldots, x_{iN}) \in R^N$ and its respective target output value $t_i \in R$.

There are two phases: feature learning phase and modeling phase.

The algorithm is as follows:

For feature learning phase:

1) Given a kernel function K and target is sat equals to input data $\mathbf{T^*=x_0}$,

2) Perform feature mapping on input data: $\mathbf{K_0 = K(x_0,x_0)}$,

3) The linear system or decision making function becomes $\mathbf{w_0K_0 = T^*=x_0}$

4) The output weight is solved by minimal least square solution: $\mathbf{w_0=K_0\dagger x_0}$

5) Update the input data or feature representation $\mathbf{x}_1 = \mathbf{w_0K_0=K_0\dagger x_0K_0}$

6) Repeat step 1, until *i* times (suppose there are *i* hidden layers)

7) The updated feature representation $\mathbf{X}_i$ for *i* hidden layer is $\mathbf{X_i=w_{i-1}K_{i-1}=K_{i-1}\dagger X_{i-1}K_{i-1}}$

Where $\mathbf{X}_i$ is the input layer for *i*+1 layer, $\mathbf{w}_{i-1}$ is the output weight of $(i\text{-}1)^{\text{th}}$ hidden layer, $\mathbf{K}_{i-1}$ is the feature mapping function for *i*-2 input layer, and $\mathbf{K}_{i-1}\dagger$ is the pseudo-inverse of a matrix $\mathbf{K}_{i-1}$.

For the modeling phase:

8) Set the target equals to the actual label as T*=T,

9) And the *i* input layer output as $\mathbf{X}_i$ to the *i*+1 layer or output layer is to be solved by minimal least square solution as $\mathbf{w=X_i\dagger T^*}$

2.1.6 ONLINE SEQUENTIAL EXTREME LEARNING MACHINES

OSELM was proposed by Liang et al. to extend ELM into online sequential learning capability [8]. OSELM consists of two phrases, namely initialization phase and sequential learning phase. The output weight matrix $\hat{\beta} = \mathbf{H}^{\dagger}\mathbf{T}$ where rank(**H**) = $\tilde{N}$ the number of hidden nodes, then the left pseudoinverse of **H**:

$$\hat{\beta} = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{T} \tag{13}$$

In the initialization phase, the appropriate matrix $\mathbf{H}_0$ is filled up by data with at least the number of hidden nodes, to make sure rank ($\mathbf{H}_0 = \widetilde{N}$). Given a set of initial training dataset $D_0 = (\mathbf{x}_i, \mathrm{t}_i),\ i = 1 \text{ to } N, D_0 \geq N$. The procedure is as follows:

1) Assign random input weights w_i and bias b_i

2) Calculate the hidden output matrix $\mathbf{H}_0$

3) Estimate the initial output weight $\beta^{(0)} = P_0 H_0^T T_0$, where $P_0 = (\mathbf{H}^{\mathbf{T}}\mathbf{H})^{-1}$

Followed by the sequential learning phase: present the $(k+1)^{th}$ chunk of new observation $D_{k+1} = \{(\mathbf{x}_i, \mathrm{t}_i)\}_{i=\left(\sum_{j=0}^{k} N_j\right)+1}^{\sum_{j=0}^{k+1} N_j}$ where D_{k+1} denotes the number of observation in $(k+1)^{th}$ chunk.

When a new data arrives, the problem then becomes minimizing: $\left\| \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix} \beta - \begin{bmatrix} \mathbf{T}_0 \\ \mathbf{T}_1 \end{bmatrix} \right\|$

Considering both chunks of training data set D_0 and D_1, then the output weight becomes

$$\boldsymbol{\beta}^{(1)} = \mathbf{K}_1^{-1} \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix}^T \begin{bmatrix} \mathbf{T}_0 \\ \mathbf{T}_1 \end{bmatrix} \tag{14}$$

$$\mathbf{K}_1 = \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix}^T \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix} \tag{15}$$

For sequential learning, $\beta^{(1)}$ should be expressed as $\boldsymbol{\beta}^{(0)}, \mathbf{K}_1, \mathbf{H}_1$ and $\mathbf{T}_1$. The formula of (1) can be written as

$$\mathbf{K}_1 = \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix}^T \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix} = \mathbf{K}_0 + \mathbf{H}_1^T \mathbf{H}_1 \tag{16}$$

Then by (16), $\beta^{(1)}$ can be written as

$$\boldsymbol{\beta}^{(1)} = \mathbf{K}_1^{-1}\begin{bmatrix}\mathbf{H}_0\\ \mathbf{H}_1\end{bmatrix}^T\begin{bmatrix}\mathbf{T}_0\\ \mathbf{T}_1\end{bmatrix} = \mathbf{K}_1^{-1}\boldsymbol{\beta}^{(0)} - \mathbf{H}_1^T\mathbf{H}_1\boldsymbol{\beta}^{(0)} + \mathbf{H}_1^T\mathbf{T}_1 \tag{17}$$

CHAPTER 3. EXPERIMENT SETUP

3.1 BENCHMARK MUSIC EMOTION DATASET

The benchmark music emotion dataset retrieved from the 2007 annual Music Information Retrieval Evaluation eXchange (MIREX) [11]. There are total 903 music divided into predefined five clusters with total of 30 emotions as showed in Table 1, each of which was judged by six IMIRSEL members. All music are used with the same format as using 16-bit sample size, 22 KHz, encoding by wav, and in 30 seconds. Then, 703 music clips of them are training data, and the remaining 200 music clips are test data.

Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**
Rowdy	Amiable/ Good natured	Literate	Witty	V olatile
Rousing		Wistful	Humorous	Fiery
Confident	Sweet	Bittersweet	Whimsical	V isceral
Boisterous	Fun	Autumnal	Wry	Aggressive
Passionate	Rollicking	Brooding	Campy	Tense/anxious
	Cheerful	Poignant	Quirky	Intense
			Silly	

Table 1. *Five emotion clusters in audio emotion classification*

3.2 FEATURE DESCRIPTION

In dynamics field, the features will be using are:

1) The frame-based RMS (mirrms), that can be calculated by

$$x_{rms} = \sqrt{\frac{1}{n}(x_1^2 + x_2^2 + \cdots + x_n^2)} \quad (1)$$

where n is the number of frame.

2) Peak of fluctuation: a fluctuation summary with its highest peak, estimating the rhythmic is based on auditory modeling on spectrogram computation transformed by auditory modeling and in each band is estimated by spectrum estimation.

3) Centroid of fluctuation: the centroid of the fluctuation summary.

In a rhythm field, the features will be using are

1) Attack time {1}: the attack times of the onset.

2) Attack time {2}: the envelope curve used for the onset detection.

3) Attack slopes: the attack slopes of the onset.

In a timbre field, the feature will be using are

1) Zero cross: The frame-decomposed zero-crossing rate.

2) Centroid of timbre: the frame-decomposed spectral centroid

3) Brightness of timbre: the fame-decomposed brightness.

4) High-frequency energy: A dual solution composed in fixing this time the cut-off frequency, and above that frequency the amount of energy is measured. The result is represented between a number of 0 and 1 , as showed in figure 1.

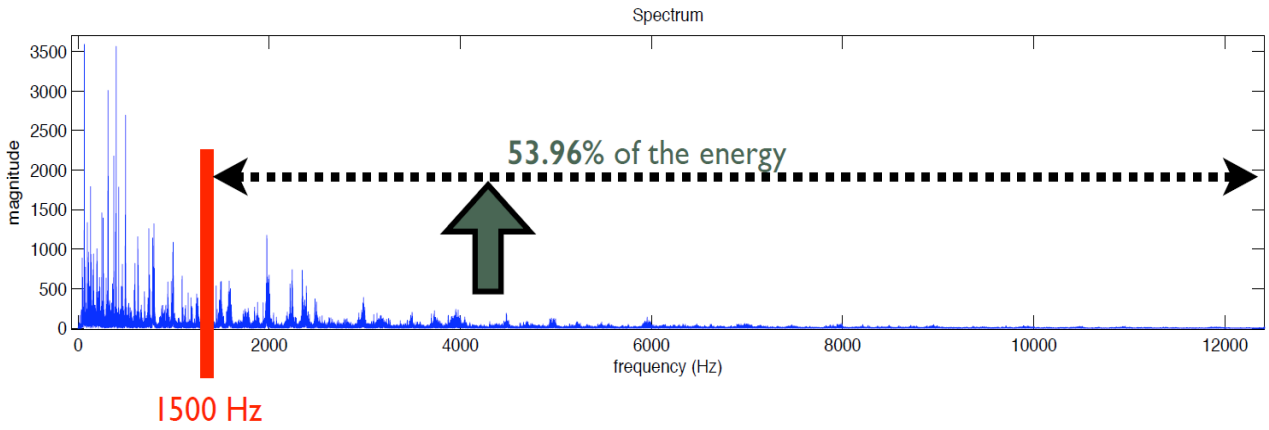


Figure 1. *The result of high-frequency energy*

5) Spread of timbre: the frame-decomposed spectral spread. The standard deviation of the data will be used. The second central moment is usually given the symbol sigma squared.

6) Skewness of timbre: the frame-decomposed spectral skewness. Skewness is the third central moment that is a measure of the symmetry of the distribution. A positive value is contained in skewness. The distribution of skewness is positively skewed with which not many values larger than the mean and thus

a long tail to right, while for the negative distribution of skew has a long tail to the left. For the symmetric distribution, it has a skewness of zero. The distribution is given by

$$\mu_3 = \int (x - \mu_1)^3 f(x) dx \tag{2}$$

where μ is the mean

7) Kurtosis of timbre: the frame-decomposed spectral kurtosis. The forth standardized moment is defined as

$$\frac{\mu_4}{\sigma^4} \tag{3}$$

Where μ is mean and σ is variance. Kurtosis is commonly defined as the forth cumulant. The probability distribution of forth cumulant is divided by the square variance, equivalent to

$$\frac{\mu_4}{\sigma^4} - 3 \tag{4}$$

which is excess kurtosis. At the end of this formula, the "minus 3" is usually describe as a correction to make the normal distribution for kurtosis = 0. For the Kurtosis of random variable, in this formula the sum is obtained by using of the cumulant, if the n independent random variables' sum is Y, all of which has the same distribution as X, then Kurt[Y] = Kurt[X]/n, but this would be very complicated if fourth standardization moment is defined for kurtosis.

8) Rolloff95 of timbre: the frame-decomposed roll-off, using a 95% threshold.

9) Rolloff85 of timbre: the frame-decomposed roll-off, using an 85% threshold.

Another method to approximate the amount of high frequency in the signal is to find the frequency which can contained a certain fraction of the total energy that below that

frequency.

As presented in figure 2, the default .85 is fixed for this ratio, other have proposed .95.

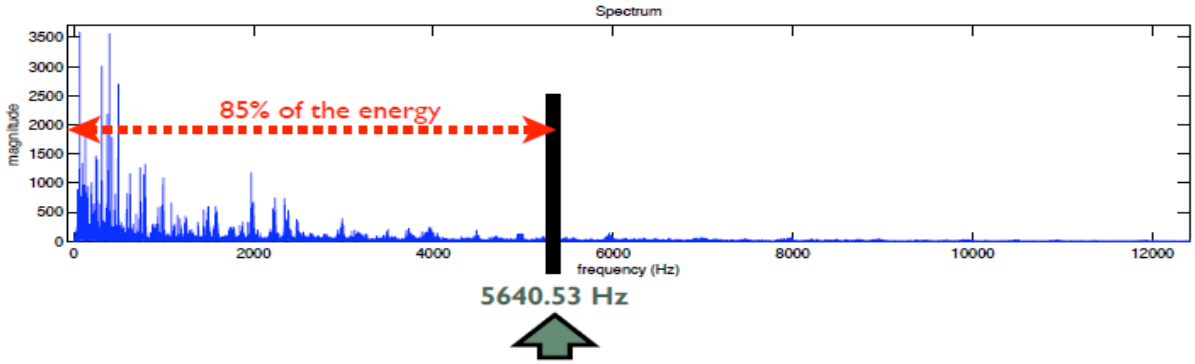


Figure 2. *The result of high frequency in the signal*

10) Spectentropy of timbre: the frame-decomposed spectral entropy. It calculates the relative Shannon (1948) entropy of the input. In information theory, the Shannon entropy is based on the following function:

$$H(X) := -\sum_{i=1}^{n} p(x_i) log_b p(x_i) \tag{5}$$

11) Flatness of timbre: the frame-decomposed spectral flatness. The flatness describes the distribution if it is smooth and get the simple ratio between the geometric mean and the arithmetic mean for the result:

$$\frac{\sqrt[N]{\prod_{n=0}^{N-1} x(n)}}{\left(\frac{\sum_{n=0}^{N-1} x(n)}{N}\right)} \tag{6}$$

12) Roughness of timbre {1}: the frame-decomposed roughness.

13) Roughness of timbre {2}: the spectrogram, containing the peaks used for the roughness estimation. The sensory dissonance or roughness is estimated by the beating phenomenon in every time the frequency closes to the pair of sinusoids. A result an estimation of roughness regardless on each pair of sinusoids the for the frequency ratio as represented in figure 3.

Figure 3. *Illustration of estimation of sensory dissonance*

Computing the peaks of the spectrum to obtain an estimation of the total roughness, and for all possible pairs of peaks all the dissonance between all possible are taken the average.

14) Irregularity of timbre {1}: the frame-decomposed irregularity

15) Irregularity of timbre {2}: the spectrogram, containing the peaks used for the irregularity estimation. For the degree of variation, the spectrum irregularity is the spectrum successive peaks.

16) Inharmonicity of timbre {1}: the frame-decomposed inharmonicity

17) Inharmonicity of timbre {2}: the spectrogram used for the inharmonicity estimation.

In inharmonicity estimation, for the amount of partials, they are not calculated by multiplying the fundamental frequency, as a value between 0 and 1. More precisely, the inharmonicity considered here takes into account the amount of energy outside the ideal harmonic series.

18) Mfcc of timbre: the frame-decomposed MFCCs.

19) Dmfcc of timbre: the frame-decomposed delta-MFCCs,

20) Ddmfcc of timbre: the frame-decomposed delta-delta-MFCCs,

Mel-frequency cepstral coefficients (MFCCs) offers a sound expression for the spectral shape. By positioning logarithmically or on the Mel scale, the frequency bands are approximated by response system of the human auditory close than the frequency bands of linear space. And the Fourier Transform is replaced by a Discrete Cosine Transform. A discrete cosine transform (DCT) is in the family of Fourier transform and is similar to the discrete Fourier transform (DFT), but real numbers is using only. It has a strong "energy compaction"

property: for most of the signal information, all concentrated information is in the DCT for a few low-frequency components. By default that is why returning the first 13 components only.

21) Low energy of timbre {1}: the low energy rate.

22) Low energy of timbre {2}: the RMS energy curve used for the low energy rate estimation;

The energy curve can be used to get an assessment of the temporal distribution of energy, in order to see if its remains constant throughout the signal, or if some frames are more contrastive than others. One way to estimate this consists in computing the low energy rate, i.e. the percentage of frames showing less-than-average energy.

23) Flux of spectral: the frame-decomposed spectral flux

Spectral flux is computed as being the distance between the spectrum of each successive frames with given spectrogram. The peaks in the curve indicate the temporal position of important contrast in the spectrogram. Fluxes are generalized to any kind of frame-decomposed representation, for instance a cepstral flux.

In the pitch field,

1) Salient of pitch: the frame-decomposed pitches.

The procedure of pitch estimation: 1) Extract pitches, 2) returned either as continuous pitch curves or as discretized note events.

2) Peak of chromagram of pitch: an unwrapped chromagram and its highest peak,

3) Centroid of chromagram of pitch: the centroid of the chromagram.

The chromagram, also called Harmonic Pitch Class Profile, shows the distribution of energy along the pitches or pitch classes.

First the spectrum is computed in the logarithmic scale, with selection of, by default, the 20 highest dB, and restriction to a certain frequency range that covers an integer number of octaves, and normalization of the audio waveform before computation of the FFT. The chromagram is a redistribution of the spectrum energy along the different pitches is presented in figure 4.

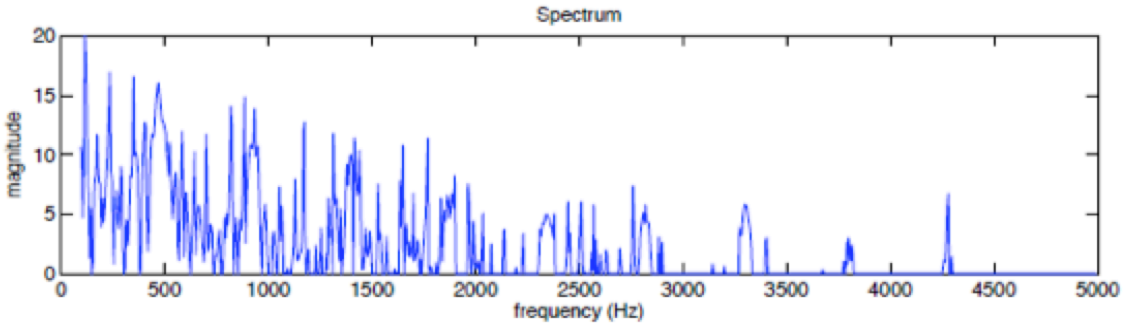


Figure 4. *The redistribution of the spectrum energy along the different pitches*

In the tonal field,

1) Key clarity of tonal: the frame-decomposed key clarity.
2) Mode of tonal: the frame-decomposed mode.

Estimates the modality, i.e. major vs. minor, returned as a numerical value between -1 and +1: the closer it is to +1, the more major the given excerpt is predicted to be, the closer the value is to -1, the more minor the excerpt might be.

3) Hcdf of tonal: the frame-decomposed HCDF.

The Harmonic Change Detection Function (HCDF) is the flux of the tonal centroid.

3.3 FEATURE EXTRACTION

In this work, for all mentioned features, they can be extracted by the MIR toolbox [12]. The features are categorized into the following four sets of features: dynamics, rhythm, spectral, and harmony, which are total of 189 features that were extracted as showed in Table 2. In dynamics field, there is RMS energy, slope and two attacks. For rhythm field, there is tempo, fluctuation peak and fluctuation centroid. For spectral field there is spectrum centroid, brightness, spread, Skewness, kurtosis, Rolloff95, Rolloff85, spectral entropy, flatness, roughness, irregularity, zero crossing rate, spectral flux, MFCC, DMFCC and DDMFCC. For the field of harmony, there is chromagram peak, chromagram centroid, key clarity, key mode and HCDF.

Set of features	Features
Dynamics	RMS energy
	Slope
	Attack
	Attack
Rhythm	Tempo
	Fluctuation peak (pos, mag)
	Fluctuation centroid
Spectral	Spectrum centroid
	Brightness
	Spread
	Skewness
	Kurtosis
	Rolloff95
	Rolloff85
	Spectral Entrophy
	Flatness
	Roughness
	Irregularity
	Zero crossing rate
	Spectral flux
	MFCC
	DMFCC
	DDMFCC
Harmony	Chromagram peak
	Chromagram centroid
	Key clarity
	Key mode
	HCDF

Table 2. *Feature sets used in this work*

3.4 PARAMETER SETTING

For feature extraction, all features are generated using MIR toolbox and sat by default. There are three different training algorithms and one update algorithm in this work. For k-NN, the number of k for the most occurred label is tested from 5 to 30. In emotion classification, the radial basis function kernel is a common choice because of its robustness and accuracy in other similar recognition tasks [6]. Therefore, SVM are trained using radial basis function (RBF) kernel, and leave other parameters unchanged. For ELM and OSELM, only the hidden code number is required and is sat from 10 to 100. The network structure of MLELM is 189-150-150-300-5 with ridge parameters 10^{-1} for layer 189-150, 10^{3} for layer 150-300 and 10^{8} for layer 300-5, and the network structure of kernel MLELM is 189-189-189-5 with regularization parameter C=1, and kernel parameter =50. Our experiments are divided into 4 experiments, which the first two explored the relationship between features sets and music emotion, and the rest updated the trained model by using OSELM, and finally compare the performance by using MLELM and kernel MLELM with all combinations of all feature sets.

Experiment 1: Four features sets are tested separately, and find a dominant features dimension.

Experiment 2: Different combinations of features sets are tested to find the best model with the best combination of features sets and classifier.

Experiment 3: After the best classifier with the best feature sets is obtained, update and test the model with modified dataset by using OSELM.

Experiment 4: Different combinations of features sets are compared with MLELM and kernel MLELM.

3.4 EXPERIMENT PROCEDURES

For experiment 1 & 2, the procedure of music emotion classification is showed as figure 5. First, a set of training data and testing data (i.e. music clips) that is annotated with music emotion and is being extracted into audio features. Then, the data can be learned by classifier

(i.e. k-NN, SVM and ELM). After a model is trained, test it with the testing data to predict the music emotion.

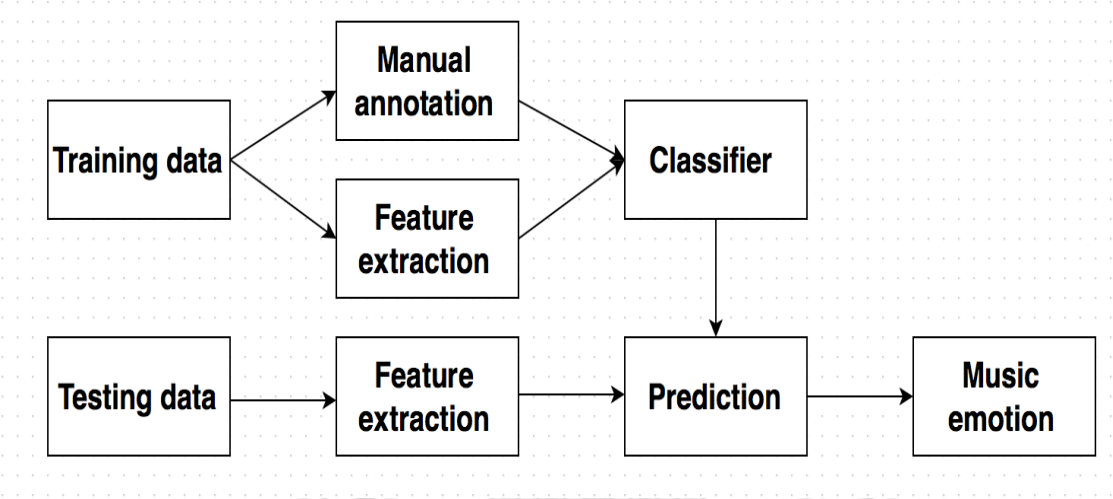


Figure 5. *The procedure of music emotion classification*

For experiment 3, after the best classifier is found, use this trained model for user customization. Given that a set of data that its class is manually modified, use OSELM to update the model as showed in figure 6.

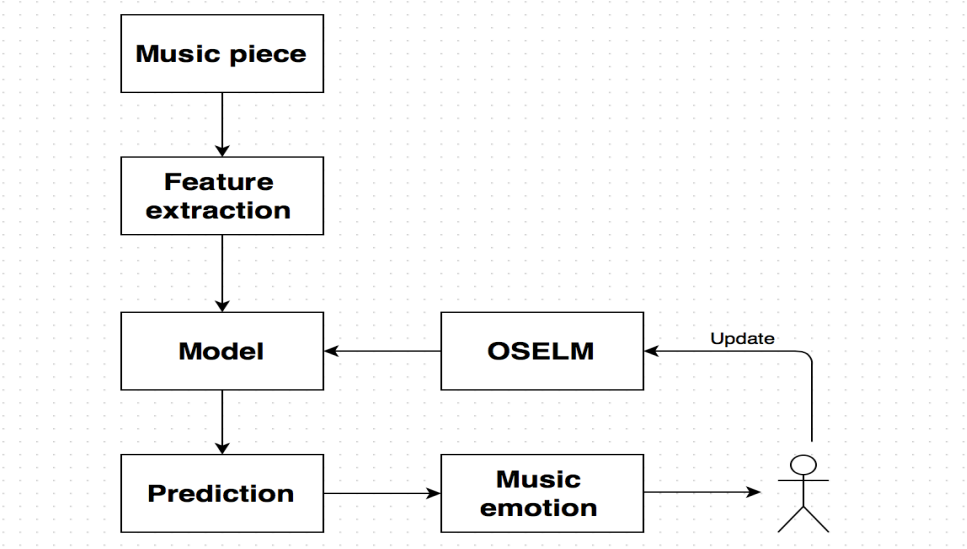


Figure 6. *The procedure of updating the model by using OSELM*

CHAPTER 4. EXPERIMENTAL RESULT

4.1 RESULT OF EXPERIMENT 1

In experiment 1, k-NN, SVM and ELM are compared with each set as shown in table 3. ELM was trained using 10 hidden nodes, and k-NN took k = 10 as the most frequent occurred label. On average, SVM and ELM are outperforming than k-NN. To our knowledge, no one has compared the features sets with ELM. However, the performance of ELM is as similar as the performance of SVM, but ELM has an advantage in multimedia processing because of its fastness that its parameters need not to be tuned. The result showed that the dominant feature set is spectral.

Table 3. *The accuracies of k-NN, SVM and ELM with each features set*

	Accuracy (%)		
Features set	*k-NN	SVM	*ELM
Dynamics	30.04	28.57	32.51
Rhythm	27.58	33.00	33.00
Spectral	29.06	34.48	**35.96**
Harmony	32.01	33.00	31.52

*ELM was trained using 10 hidden nodes, and *k*-NN takes *k* = 10 as the most frequent occurred label

4.2 RESULT OF EXPERIMENT 2

In experiment 2, in order to choose the best model, the different combination of four features sets are compared as presented in table 4. ELM was trained using 10 hidden nodes, and *k*-NN took *k* = 10 as the most frequent occurred label. Similar to experiment 1, SVM and ELM are outperforming than k-NN, and the performance of ELM is as similar as the performance of SVM. However, due to the fastness of ELM, ELM is considered with the better performance than SVM. The best combination of features sets with ELM is the use of Dynamics, rhythm and harmony features set. The result showed that not all features are useful in music emotion

classification, and the usefulness of the combination of features sets are based on the use of different machine learning algorithms. For example, the best features sets combination for SVM is both the use of all features sets and the use of rhythm, spectral and harmony.

Table 4. *The accuracies of k-NN, SVM and ELM with different combination of feature sets*

	Accuracy		
Feature set	**k*-NN	SVM	*ELM
Dyn + Rhy	28.57	32.02	34.98
Dyn + Spec	29.56	34.98	27.09
Dyn + Har	32.01	34.48	34.48
Rhy + Spec	29.56	35.96	30.54
Rhy + Har	31.52	36.45	32.51
Spec + Har	31.52	36.45	24.63
Dyn + Rhy + Spec	28.57	34.48	28.57
Dyn + Rhy + Har	32.51	34.98	**36.52**
Dyn + Spec + Har	32.02	35.47	33.54
Rhy + Spec + Har	33.50	**36.95**	30.07
All sets	30.04	**36.95**	29.56

*ELM was trained using 10 hidden nodes, and *k*-NN takes $k = 10$ as the most frequent occurred label

4.3 RESULT OF EXPERIMENT 3

ELM is considered providing the best performance with the best combination of feature sets that are dynamics, rhythm and harmony. Then, when the label of music piece is modified, OSELM is employed to update the model one by one.

OSELM inherits the fastness of ELM that the parameters need not to be turned but with similar accuracy, so that it is appeal for the model update. In experiment 3, OSELM takes the

origin music emotion dataset in its initial learning phase, and update the model in its sequential learning phase. Given an origin dataset with 10%, 20%, 30%, and 40% random modification of its label, the model updated sequentially by using OSELM as shown in table 5. The result suggested that using OSELM to update model remained similar accuracy with fast update speed.

Table 5. *The training time and accuracies of updating an origin dataset by using OSELM*

	*OSELM	
Data Modified (%)	Training Time (s)	Accuracy (%)
10	0	36.10
20	0	35.44
30	0.0156	36.77
40	0.0156	35.77

* OSELM was trained using 20 hidden nodes and using 1 chunk size, and update each data one by one

4.4 RESULT OF EXPERIMENT 4

As presented in table 6, for most of the combinations of feature sets, kernel MLELM performs better than MLELM, for both in accuracy and training time. The reason would be the number of layers used in both algorithms. kernel MLELM uses two layers but MLELM uses 3 layers, however, for the performance kernel MLELM is better. This means that the learning ability for *k*-MLELM is better than MLELM. For both algorithms, the best result is obtained by using all features sets. This means that for using more features, the deep learning method can extract more information from those features than the shadow learning method. MLELM and kernel MLELM are outperform than *k*-NN, SVM, and ELM. The best performance is by using kernel MLELM with three features sets that are rhythm, spectrum and harmony and with all features sets.

Table 6. *Time training time and accuracies of using MLELM and kernel MLELM*

	*MLELM		*kernel MLELM	
Feature set	Accuracy(%)	Training time (s)	Accuracy(%)	Training time(s)
Dyn + Rhy	30.54	0.09	**31.03**	**0.04**
Dyn + Spec	37.93	0.1	**42.86**	**0.06**
Dyn + Har	**34.48**	0.09	33.50	**0.04**
Rhy + Spec	38.92	0.09	**42.36**	**0.05**
Rhy + Har	**37.44**	0.08	36.95	**0.04**
Spec + Har	41.87	0.1	**44.83**	**0.06**
Dyn + Rhy + Spec	43.84	0.09	**44.83**	**0.06**
		21		
Dyn + Rhy + Har	39.90	0.08	**36.95**	**0.04**
Dyn + Spec + Har	43.35	0.1	**45.32**	**0.05**
Rhy + Spec + Har	43.84	0.1	**46.80**	**0.06**
All features	45.32	0.1	**46.80**	**0.06**

*The network structure of MLELM is 189-150-150-300-5 with ridge parameters 10^{-1} for layer 189-150, 10^{3} for layer 150-300 and 10^{8} for layer 300-5, and the network structure of kernel MLELM is 189-189-189-5 with regularization parameter C=1, and kernel parameter =50.

4.5 DISCUSSION

The benchmark music emotion dataset is retrieved from the 2007 annual Music Information Retrieval Evaluation eXchange (MIREX). However, it only contained 903 music with annotation of music emotion. This sample size may not produce an accurate result in classification. However, constructing a large music emotion dataset is expensive and very time-consuming, and no alternative online dataset is found. Moreover, Although OSELM can obtain a similar accuracy with very fast speed, OSELM performs different results in each run because its performance depends partially on the randomly generated parameters i.e. $(\boldsymbol{\alpha}_i, b_i)$.

In giving a music recommendation, Music listener is likely to request the update of the model one by one. Thus, in this work the chunk size for update is sat to 1 and the suboptimal result is by running OSELM several to hundreds epochs. For kernel MLELM, using two layers multi layer extreme learning machines can obtain a good performance compared to using three layers for MLELM. For MLELM, tuning the parameters for each layer is exhaustive, but for kernel MLELM, fewer parameters are required to tune. However, kernel MLELM requires more memory than MLELM if using the same number of layer because of using kernel.

CHAPTER 5. CONCLUSION

In this work, traditional machine learning as k-NN, and state-of-art neural network as SVM and ELM are applied and compared with different combination of four features sets in music emotion classification. Four features sets are dynamics, rhythm, spectral and harmony, all of which are extracted from matlab toolbox called MIR toolbox. And the training and testing dataset are retrieved from the 2007 annual Music Information Retrieval Evaluation eXchange. The performance of k-NN is the worst and the performance of SVM and ELM are similar. However, ELM is considered providing the best result because of its fastness property that it can randomly generate its parameters. The best performance of ELM with the best combination of features sets is the use of dynamics, rhythm, and harmony, which suggested that not all features sets are useful in music emotion classification. However, the usefulness of different combination of features sets depends on the applied machine-learning algorithm. For example, the best combination of features sets for SVM is different from ELM's. In user customization, OSELM is proposed to update the model based on user's preference. Experimental result showed that after the model updated by using OSELM, the accuracy remained similar with very fast speed.

For deep learning methods such as MLELM, and kernel MLELM, the performance for music emotion task is outperform than any shadow learning methods such as *k*-NN, SVM, and ELM. In addition, MLELM, and *k*-MLELM can fully employ the theory of ELM which once the parameters are randomly generated for each layer, no fine tuning is required unlike the traditional deep learning methods. This significantly increases the training time for deep networks by using ELM-based deep networks. In this work, the best performance in this emotion classification task is by kernel MLELM with all features sets, or with three features sets that are spectral, harmony, and rhythm.

CHAPTER 6. ETHICS AND PROFESSOR CONDUCT

In this project, we gave the proper credit for intellectual property by citation such as using inline citation like [1] and gave all the references at the end of the report. By doing do, we did not take credit for other's idea and work. We also honored property rights including copyrights of other's work. For example, when we using a free matlab toolbox like libsvm, we follows the required usage of the toolbox by citating the toolbox at the end of the report.

BIBLIOGRAPHY

[1] J.J. Deng and C. Leung, Emotion-based Music Recommendation Using Audio Features and User Playlist. In *Proceedings of the 6th International Conference on New Trends in*, pages 796-801, 2012

[2] T. Magno and C. Sable. A Comparison of Signal of Signal-based Music Recommendation to Genre Labels, Collaborative Filtering, Musicological Analysis, Hu- man Recommendation and Random Baseline. In *Proceedings of the 9th International Conference of Music Information Retrieval*, pages 161–166, 2008.

[3] K. Hoashi, K. Matsumoto, and N. Inoue, "Personalization of user profiles for content-based music retrieval based on relevance feed- back," *ACM Multimedia*, 2003, pp.110–119.

[4] HORSBURGH, B., CRAW, S. and MASSIE, S., 2015. Learning pseudo-tags to augment sparse tagging in hybrid music recommender systems. Artificial Intelligence, 219, pp. 25-39.

[5] T.L. Pao and Y.M. Cheng, Comparison between Weighted D-KNN and Other Classifiers for Music Emotion Recognition, In *Proceedings of the 3rd International Conference on Innovative Computing Information and Control*, page 530, 2008

[6] Y. Music, S. Dixon, and M. Pearce, "Evaluation of musical features for emotion classification," in Proc. ISMIR, Oct. 2012.

[7] S. Scardapane, D. Comminiello, M. Scarpiniti, and A. Uncini, "Music classification using extreme learning machines," in *Proc. 8th Int. Symp. Image Signal Process. Anal. (ISPA)*, Trieste, Italy, Sep. 2013, pp. 377–381.

[8] Lan, Y, Soh, YC, Huang, GB (2009) Ensemble of online sequential extreme learning machine. Neurocomputing 72: pp. 3391-3395

[9] B. Shao, M. Ogihara, D. Wang, T. Li, Music recommendation based on acoustic features and user access patterns IEEE Transactions on Audio, Speech, and Language Processing, 17 (2009), pp. 1602–1611

[10] Banerjee, K.S.: Generalized Inverse of Matrices and Its Applications. Technometrics. 15, 197-197 (1973)

[11] X. Hu, J.S. Downie, C. Laurier, and M. Bay. The 2007 MIREX Audio Mood Classification Task: Lesson Learned. In *International Society for Music Infor- mation Retrieval Conference*, pages 462–467, 2008.

[12] O. Lartillot and P. Toiviainen. MIR in Matlab (II): A Toolbox for Musical Feature Extraction from Audio. In *International Conference on Music Information Retrieval*, number Ii, pages 237–244, 2007.

[13] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for supportvector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[14] O. Celma and P. Lamere. Tutorial on music recommendation. Eight International Conference on Music Information Retrieval: ISMIR 2007; Vienna, Austria, 2007.

[15] O. Celma, Music Recommendation and Recovery The Long Tail, Long Fail, and Long Play in the Digital Music Space, Springer, 2010, page 1-3.

[16] M. Muller, Information for Music and Motion, Springer, 2007

[17] X. Su and T.M. Khoshgoftaar, A Survey of Collaborative Filtering Techniques. ACM Transaction on Advances in Intelligent Systems, 2009

[18] G. Tzanetakis and P. Cook, “Multifeature audio segmentation for browsing and annotation,” in Proceedings of the 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 1999.

[19] A. Rauber E. Pampalk and D. Merkl,“Content-based organization and visualization of music archives,” in Proceedings of the 10th ACM International Conference on Multimedia, 2002, pp. 570–579.

[20] L. Rabiner and B.H. Juangl, Fundamentals of Speech Recognition, Prentice-Hall, 1993.

[21] G. Tzanetakis and P. Cook, “Multifeature audio segmentation for browsing and annotation,” in Proceedings of the 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 1999.

[22] O. Lartillot and P. Toiviainen, A Matlab Toolbox for Musical Feature Extraction from Audio, on the Proceeding of 10th International Conference on Digital Audio Effects, 2007

[23] X. Hu, J.S. Downie, and A.F. Ehmann. Lyric Text Mining in Music Mood Classification. In 10th International Society for Music Information Retrieval Conference, number Ismir, pages 411–416, 2009.

[24] J.Y. Lee, J.Y. Kim and H.G. Kim, Music Emotion Classification Based on Music Highlight Detection, 2014 International Conference on Information Science and Applications, 2014

[25] K. Markov, M. Iwata, and T. Matsui. Music emotion recognition using gaussian processes. In Proceedings of the ACM multimedia 2013 workshop on Crowdsourcing for Multimedia, CrowdMM. ACM, 2013.

[26] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, “Extreme learning machine for regression and multiclass classification,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 42, no. 2, pp. 513–29, 2012.

[27] Vapnik, V. (1995). "Support-vector networks". Machine Learning 20 (3): 273. doi:10.1007/BF00994018

[28] Altman, N. S. (1992). "An introduction to kernel and nearest-neighbor nonparametric regression". The American Statistician 46 (3): 175–185. doi:10.1080/00031305.1992.10475879

[29] L.L.C. Kasun, H. Zhou, G.B. Huang & C.M. Vong,”Representational Learning with Extreme Learning Machine for Big Data”, in IEEE Intelligent System, 2013