Exploring the General Melodic Characteristics of XinTianYou Folk Songs

Juan Li[+] **Lu Dong*** **Jianhang Ding*** **Xinyu Yang***

{ [+]Centre for Music Education, *Department of Computer Science & Technology}, Xi'an Jiaotong University

Emails: lijuan@mail.xjtu.edu.cn, {donglu666, jh.ding}@stu.xjtu.edu.cn, yxyphd@mail.xjtu.edu.cn

ABSTRACT

This paper aims to analyze one style of Chinese traditional folk song named Shaanxi XinTianYou. By analyzing the melody of this folksong genre, we make a clear, vivid, and thus easily approachable presentation of the cultural characteristics and significance of XinTianYou. Comparing to previous researches which mainly focus on mathematics and statistics, we further consider the musical continuity. Our insight is that, the combination of intervals reflects the characteristics of the music style. The significant pattern of the combinations can be used as representations of XinTianYou. We build a MIDI database, based on which the most representative combination of intervals are extracted. We propose to use N-Apriori algorithm which counts the frequent patterns of melody. Considering both the significance and similarity between music pieces, we provide a multi-layer melody perception clustering algorithm which uses both the melodic direction and the melodic value. The experiment results are analyzed based on both pattern mining techniques and music theories. For evaluation, we asked experts in this field to mark our results and proved that our results are consistent with the expert's intuition.

1. INTRODUCTION

XinTianYou is a style of improvised folk song in mountain area of Shaanxi province in China. It is the most important component of Shaanxi folk songs, and quite unique and attractive. Explortion on XinTianYou is beneficial to the understanding of this music style and also the applications such as music information retrieval. Moreover, it facilitates musical education and composition in Xintianyou as well as contributes to its inheritance, protection, and development.

With the development of music digitalization, machine learning techniques have been widely used for music information analysis and greatly promoted the music research. Naoko kosugi.et.al [1] built a music dataset and developed a retrieval system. They compared the similarity between music pieces based on rhythm information. Li.et.al [2] improved this system by taking the pitch and rhythm into consideration. They proposed to do matching based on the geometrical similarity of melodic contours. Yang.et.al [3] investigated mood categories with audio features, and used it for comparison between Chinese music and western music. These research studies have proven to be effective for music retrial and music emotion analysis, but mainly focus on the differences between music genres rather than the general characteristics within a genre. In this paper we focus on research of general characteristics for one genre because it is potentially useful for the music composition and understanding of the development route of music as well.

In addition, most previous researches focus on single feature and the statistical analysis of the feature. For example, the Alicante set [4] contains 28 global features based on the statistics of pitch, duration and the mean or standard deviation. The Mckay set [5] contains 109 global features, based on device, texture, rhythm, dynamic, pitch statistics, melody and chords. These studies only focus on mathematics and statistics, and ignore the musical continuity.

Some patterns mining algorithms have been introduced in [6]. In this paper, a new perspective based on the combination of interval is proposed to consider the integrity and continuity of music. An interval reflects the relative tendency of the two pitches. The combination of intervals reflects the trend of a melody. We use the N-Apriori algorithm to mine for frequent interval combinations. Additionally, based on the redundancy-aware top-*k* idea, we further use a multi-layer melody perceptive cluster algorithm to cluster the frequent interval combinations. Then the significant patterns are selected as the general characteristics of XinTianYou.

The organization of the paper is as follows. We first explain the preprocessing of data in Section 2. Section 3 contains the description of the N-Apriori patterns mining algorithm. Next we present how to use the multi-layer melodic perception to do the clustering in Section 4. The experimental results are shown in Section 5. Conclusions and future research are in Section 6.

2. PREPROCESSING

We build a XinTianYou MIDI database for all the XinTianYou songs collected by [7]. We choose this format because features can be easily computed. Each MIDI file only contains one repeat of melody to avoid redundancy, and no ornament is added to the main melody. After these initial preparations, we first divide each piece from the database into segments, and then map the difference between adjacent pitches to the interval.

2.1 Melody Segmentation

Traditional Chinese music is normally the combination of short units. Some Chinese researchers have investigated the short melody units. Liu [8] reported a tricolor-theory about restricting the traditional style of music. In his article, several sequences of three notes within a 4th were found and named musical chromosomes. Wang [9, 10] observed that Chinese folk music includes five phonological systems and each of them is a specific interval structure composed of three or four notes. Based on these studies, we believe it is important to extract general characteristics from the segments of XinTianYou melodies.

To extract the short melody units, we need to divide the melody into segments based on the semantics of the lyrics, because lyrics and tunes are depend on each other in Chinese folk songs. Most XinTianYou songs contain eight measures which correspond to two sentences. Generally, two measures express as a semantic element. Also we find that normally the singer needs a pause to breathe after singing two or three measures. Based on these observations, we first divide the music piece into segments, each of which contains two measures. Then we manually mark the sentences in the database to make sure the division is within the range of each sentence. The last measure is combined with the one before it when there is an odd number of measures in one sentence. Our method can divide the music piece to segments reasonably. In Figure 1 we show the division of an example song.

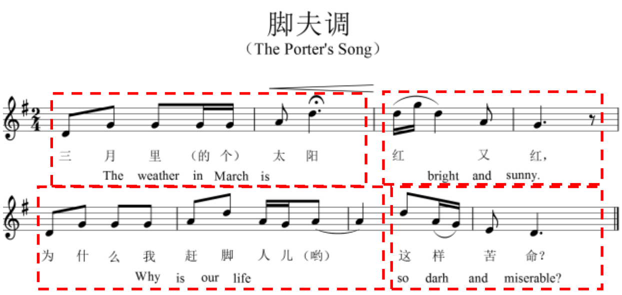


Figure 1. Segmentation of an example song. Each dashed box contains a segment. The lyrics reflect the laboring people's complaining about their miserable life and yearning for happiness. The song can be sung repeatedly with this fix tune structure.

2.2 Mapping

The MIDI data in our database is extracted by using the open source MIDI Toolbox [11]. Notes are represented by hexadecimal numbers from the 00 to 7F. The difference in numerical values between notes corresponds to the number of semitones. Under the naturally symmetric intervals, there is a certain relationship between the number of semitones and the interval distance (without considering the interval property). This relationship appears in all octaves. Next, we summarize the mapping in two cases.

Case 1: Mapping within an octave.

We show the results in Table 1. Here we use "*Diff*" to represent the difference between the two adjacent notes in numerical values and "*Interval*" to represent the corresponding musical distance.

As we can see from Table 1, when the *Diff* is six, there are two options. This is because the augmented 4th and diminished 5th are naturally symmetric intervals. If two notes are from the same register, for example f^1 and b^1, the *Interval* is 4th. But if the two notes are from adjacent registers, for example $b^1$and f^2, the *Interval* is 5th.

Diff	**0**	**1, 2**	**3, 4**	**5, 6**	**6, 7**	**8, 9**	**10,11**	**12**
Interval	unison	2nd	3rd	4th	5th	6th	7th	octave

Table 1. Relationship between *Diff* and *Interval*.

Case 2: Mapping beyond one octave (*Diff>12*).

The interval can be computed by the following Equation (1):

$$\begin{cases} Interval = \mathrm{F}(X_1) + X_2 \times 7 \\ X_1 = \mathrm{Mod}(Diff, 12) \\ X_2 = \mathrm{Fix}(Diff / 12) \end{cases} \tag{1}$$

where the function Mod() is modulo arithmetic, and the Function Fix() is quotient operation. F() is a discrete function which maps the semitones to interval within one octave, as shown in Table 1. In this way, we can turn MIDI data into musical intervals.

3. N-APRIORI PATTERN MINING

In this section, we introduce an improved Neighbor-Apriori (N-Apriori) algorithm to explore the frequent patterns. We mainly improve the "join step" and the "search step" of the traditional Apriori algorithm. Algorithm 1 shows pseudo-code for the N-Apriori algorithm.

Algorithm 1. N-Apriori algorithm.

Input *D*: Database.
Output $\{L\}$: Frequent items.

1. Find frequent_1_items L_1 ;
2. **The join step** : candidate *k*-items generation C_k,
3. $C_k = L_{k-1} \times L_1$;
4. **The search step**: Candidate *k*-items adjacent statistics;
5. **The prune step** : frequent *k*-items generation L_k, $C_k \rightarrow L_k$;
6. Repeat 2~5, until there are no more frequent items remaining;
7. Return frequent items $\{L\}$;

Here we explain the process of N-Appriori in detail. Firstly, same as Apriori algorithm in [12], the N-Appriori algorithm scans each transaction in the database, and counts the frequency of each item. Here, a transaction refers to a semantic fragment, and item refers to the interval calculated by the Equation (1). A pattern could be one item or combination of several items. The frequency of each pattern is referred as support. Any pattern with support lower than a minimum support threshold is removed. N-Apriori also employs an iterative approach known as a level-wise search, where *k*-items are used to explore (*k+1*)-items. However, in the join step, the set of candidate k-item C_k is generated by joining L_{k-1} to L_1, and the identical items are not merged. In the search step, each pattern in C_k is regarded as a whole. In our method, only the patterns with specific context and order are counted, while in the original Apriori algorithm all patterns are considered. These processes are repeated until no more frequent items can be found.

4. MULTI-LAYER MELODY CLUSTERING

Although by using the N-Apriori algorithm we can find correct and reasonable intervals, it is not clear how to find an ideal support threshold. If the threshold is too low, it may lead to a large number of output patterns, while a high threshold may fail to find representative modes. In order to compress the number of frequent patterns and find high-quality melodic framework, Dong et al [13] propose to extract redundancy-aware top-*k* patterns from a large collection of frequent patterns. Their method uses significance and redundancy as criterions.

Inspired by [13], we propose to use a multi-layer melody perception clustering method for extracting top-*k* representative patterns. We use the Cosine Similarity to determine the similarity between melodic contours, and Edit Distance to determine the similarity of melodic amplitude. In order to extract high significance and low redundancy melodic structures from the combinations of intervals, we use Hierarchical Agglomerative Clustering (HAC) method, which has been commonly used for document similarity clustering [19-21]. We apply HAC to the similarity or cost matrix computed by Cosine Similarity and Edit Distance. To evaluate the clustering results, we use the average Silhouette Coefficient (SC) [14] and find the best number of clusters. At last the most significant patterns are selected as the general characteristics of XinTianYou.

4.1 Melodic Direction based Clustering

The tendency of a melodic line includes ascending and descending stages, which can be represented as its melody contour. So the problem of clustering the melodic direction can be converted to the comparison between contours. Cosine Similarity (CS) has been commonly used to solve the problem of matching feature vectors [15-17]. It can be used to solve the problem of matching melodic contours. But the framework of N-Apriori algorithm form patterns of different sizes, between which the CS method cannot be directly applied. For the vectors with same length, the CS can be directly computed. For the vectors with different sizes, we calculate the CS between the shorter vector and each sub-vectors of the longer vector, and get a list of corresponding similarity values. We then compute the mean of all the similarity values as the final similarity. After computing the similarity matrix, the hierarchical clustering method can be applied. The pseudo-code is given in Algorithm 2.

Algorithm 2. Direction based clustering algorithm.

Input L: Frequent pattern sets $L=\{L_1,L_2...L_n\}$.
Output DCluster: Clustering Result

```
CosValue = 0;      //Cosine value
MDirection = ∅;  //Similarity matrix
For  i = 1: Length(L)
   For  j = i : Length(L)
      If  length(Li) == length(Lj)  CosValue = Cos(Li, Lj);
      Else  CosValue = mean(Cos);
                    //Take the average of all cosine values
      EndIf
   MDirection [i, j] = CosValue;
   EndFor
EndFor
DCluster = HAC_Cluster(MDirection);
                 //Clustering based on the similarity matrix
Return DCluster;
```

4.2 Melodic Value based Clustering

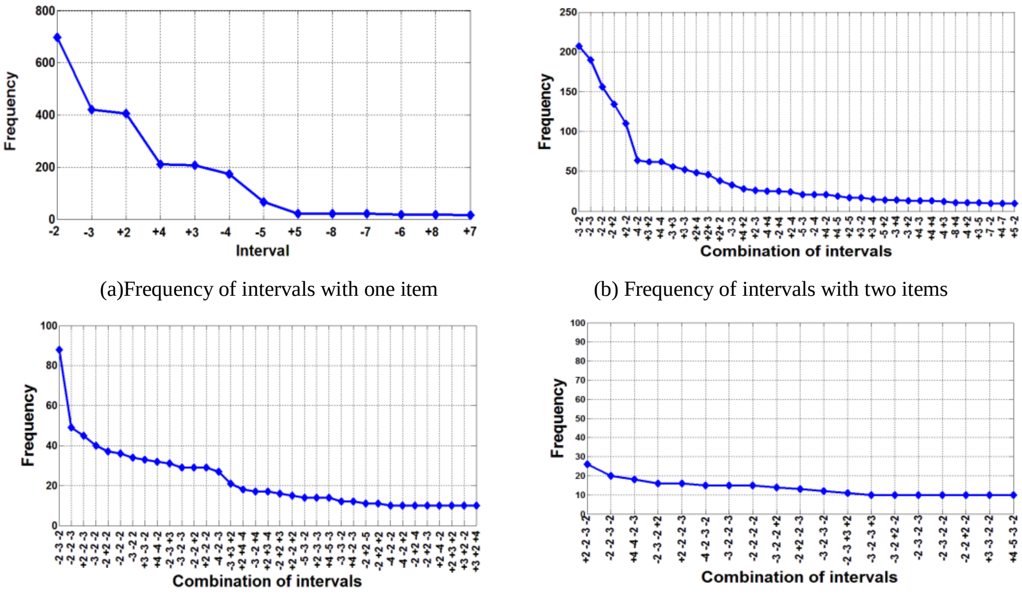
The values within the combination of intervals represent the amplitude of the melody in a piece of music. In this section we apply Edit Distance (ED) [18] to compare the amplitude values of different melodic lines. The basic idea of ED is to measure the cost of transforming one string to another through a dynamic programming process. The typical transformations include substitution, insertion and deletion. We propose an improved ED method named self-adaptation cost Edit Distance (SAC ED) to compute the distance between two vectors. When an interval ***a*** is inserted or deleted, the cost of the transformation is $|\boldsymbol{a}|$, which represents the norm of ***a***. If ***a*** is replaced by ***b***, the cost is the norm of the difference between the two vectors $|\boldsymbol{a}-\boldsymbol{b}|$. Therefore the distance function can be summarized by equation (2).

$$Cost = \begin{cases} |\boldsymbol{a}| \text{ if } \boldsymbol{a} \text{ is insersted} & (InsCost) \\ |\boldsymbol{a}-\boldsymbol{b}| \text{ if } \boldsymbol{a} \text{ is replaced by } \boldsymbol{b} & (RepCost) \\ |\boldsymbol{a}| \text{ if } \boldsymbol{a} \text{ is deleted} & (DelCost) \end{cases} \tag{2}$$

Algorithm 3. Computing self-adaptation cost Edit Distance (SAC ED).

Input String *A* and *B*: frequent patterns in each *Dcluster*.
Output *DistValue*: The minimum cost between *A* and *B*.

```
DistMatrix = ∅;
La = Length(A);
Lb = Length(B);
For  p = 0 : La-1
   For  q = 0 : Lb-1
      If p == 0 && q==0         DistMatrix (p, q) = 0;
      Elseif  p == 0 && q > 0     DistMatrit(p, q) = InsCost;
      Elseif  p > 0 && q == 0     DistMatrix (p, q) = DelCost;
      Elseif  P > 1 && q > 1
                 Ins = DistMatrix(p-1, q) + InsCos;
                 Rep = DistMatrix (p-1, q-1) + RepCost;
                 Del = DistMatrix (p, q-1) + DelCost;
           DistMatrix (p,q) = Min{ Ins , Rep , Del };
      EndIf
   EndFor
EndFor
DistValue = DistMatrix (La-1, Lb-1);
Return  DistValue;
```



(a)Frequency of intervals with one item

(b) Frequency of intervals with two items

(c)Frequency of intervals with three items

(d) Frequency of intervals with four items

Figure 2. Frequency sets computed by N-Apriori

The transformation between two interval vectors is not unique, so there are different distance values. Among these values we choose the minimum one to measure the similarity between two interval vectors. The interval vectors are more similar if the distance value is lower. Computing the minimum SAC ED is a dynamic programming process, which is shown by pseudo-code in Algorithm 3. With SAC ED method, we can compute a cost matrix, based on which we can do the HAC clustering similar to Algorithm 2.

4.3 Evaluation of Clustering Results and Significance Analysis

To evaluate the clustering results, we compute the average Silhouette Coefficient (SC) [14] value of all objects in the cluster. The range of average SC values is [-1, 1]. The larger the average SC is, the higher the clustering quality will be.

After the clustering, each cluster is relatively independent, but the coherence is high within each cluster. So the most significant pattern is selected from each cluster as the general characteristic of melodies of XinTianYou. In our experiment, we use the “support” introduced in Section 3 to measure the significance of pattern. More specifically, we choose the one with highest support value as the most significant pattern.

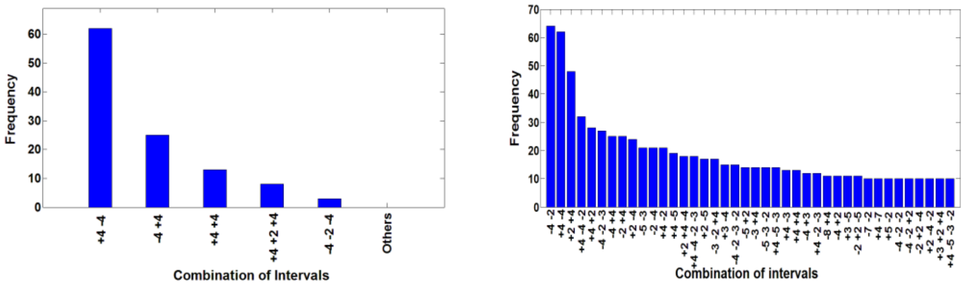
5. EXPERIMENTS

5.1 Results of N-Apriori

In total we extracted 453 semantic fragments from the 109 songs in our database. The N-Apriori method is applied to these fragments. The support threshold is set to 10 in our experiment. The resulting frequency is shown in Figure 2. The four graphs represent the frequency pattern of intervals with different lengths. The horizontal axis indicates different interval patterns. The sign '+' and '-' in the axis notes denote the melody ascending and descending respectively. The vertical axis indicates the corresponding frequency. Interval of unison is not shown in the results. This is because unison represents two adjacent notes with the same pitch, which cause no changing in the melody trend. The adjacent intervals are merged in our experiment.

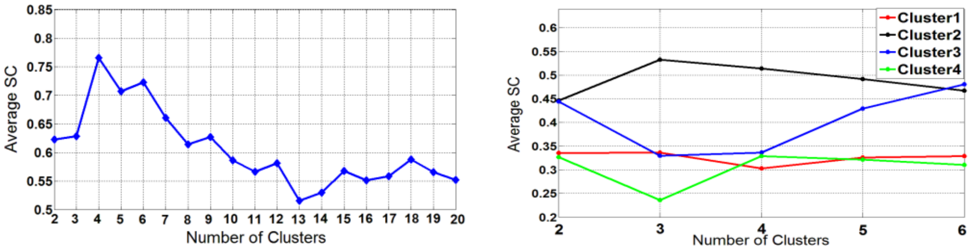
From Figure 2-(a) we can see that frequency sets within the interval from -2 to +2 take a large proportion, and the trend slows down from +4 to -4. In Chinese folk songs, intervals no more than 3rd are regarded as narrow intervals [22], and intervals above 3rd are regarded as wide intervals [22]. Usually the narrow intervals appear frequently, however, 4th is dominant in our experiment, which appears around 400 times and takes about 88% of the semantic fragments. Figure 2-(b) shows that for combination of a 2nd and a 3rd, the descending trend appears to be more prominent than the ascending trend. We can also see that symmetric structure appears a lot in the high proportion combinations, such as +2 -2,-2 +2, +3 -3, 3 +3, +4 -4. Figure 2-(c) shows that the structure -2-3-2, is much higher than the second frequent structure -2-2-3. They tend to be evenly distributed after adding 4th. Figure 2-(d) presents an obvious decline in frequency compared to the three previous results. The distribution is relatively stable at the low frequency. Next we analyze the result from the aspects of the direction trend, pitch selection, and musical structure.

Regarding the direction trend, a narrow structure with a continuous descending is dominant in XinTianYou, while continuous ascending structures rarely happen. The frequency of -3-2 and -2-3 structures verify the "three notes within a 4th" structure theory proposed by Liu [8]. In this paper they regarded the combination of 2nd and 3rd from the same direction as being the chromosome of Chinese folk songs. Moreover, the results show that the chromo



(a)Analysis of double-4th distance (b) Frequency of wide intervals

Figure 3. Deep analysis of wide intervals



(a)Average SC results based on CS (b)Average SC results based on ED

Figure 4. Average SC results

some - the top frequent patterns - also includes other combinations of "2nd and 3rd" and "2nd and 2nd".

In the aspect of pitch selection, earlier researchers considered that a notable feature of XinTianYou is the "double-4th" structure which usually in two forms of +4+2+4 and +4+4 [7]. But according to the results in Figure 3, the main 'double-4th' structures are +4-4 and -4+4, rather than +4+2+4 and +4+4 (as shown in Figure 3-(a)). We also found the wide interval frequently appears in various combinations as shown in Figure 3-(b). This makes XinTianYou full of strength and vitality.

Regarding the structure, we found that the symmetric structures are extensively used. This introduces balance into the melody, and adds flexibility without losing harmony. This enables the music to present a sense of destination seeking and provoke deep feelings in the heart. Long melodic lines have various patterns, as they are a flexible mixture of small sections, for which the general characteristics is not very obvious.

5.2 Results of Multi-layer Clustering

In this section, based on the frequency analysis by using the N-Apriori algorithm, we first use the Algorithm 2, and then use Algorithm 3 to do the clustering. Then the average Silhouette Coefficient (SC) is used to evaluate the clustering results.

Figure 4-(a) shows the average SC results under different cluster numbers. It can be seen that using 4 clusters leads to the highest average SC value. So we use 4 clusters in this stage. In Figure 4-(b), we show the average SC for the further clustering of each cluster from last step. The best selected number from cluster 1 to cluster 4 is 3, 3, 6 and 4. So we obtain 16 clusters in total. For each of these 16 clusters, the combination of intervals with the highest support is selected by using the top-*k* method. All these representative combinations form the final general characteristics of XinTianYou. The results show in Table 2. The corresponding melodies are shown in the Appendix.

Cluster	Interval Combination	Cluster	Interval Combination
k=1	-3 -2	k=9	+4 +4
k=2	-2 -3 +3	k=10	+3 +2 +4
k=3	-7 -2	k=11	+3 +2 -2
k=4	-2 +2 -2	k=12	+2 +4 -4
k=5	-3 +3	k=13	+2 -2 -3
k=6	-8 +4	k=14	+4 -4
k=7	+2 +3 +2	k=15	+4 -7
k=8	+2 +4	k=16	+4 -4 -2 -3

Table 2. XinTianYou's general characteristics in melody

To evaluate our results, we asked ten experts to manually mark the resulting musical data. All the participants are professionals in music and familiar with XinTianYou. More specifically, we first choose the most frequent melody segments for each of the 16 clusters and built the first set of data "Group One", and then randomly choose 16 other melodies appear in our database named it "Group Two". The Group One and Group Two are put together. Table 3 shows the melody scores for the 32 melody segments. All these melody segment scores are shown to the participants in random order. The participants were asked to mark each melody to one of five levels which represent how frequent the melody appears in XinTianYou. Level 1 means not frequent at all, while the level 5 means the most frequent. In Figure 5, we visualize the marks of both the 16 representative melody segments found by our method (in blue) and the randomly chosen noisy data (in red). The horizontal axis lists the segment id and the vertical axis indicates the corresponding mark. The item id in Table 3 is in the same order as the horizontal axis of chart in Figure 5. We also show the average mark for each group by the black dashed lines. From the chart we can see that obviously the 16 melody segments found by our method obtains a much higher mark (4 on average) while the noisy data (Group Two) only get mark 1.75 on average. This means the experts think that the 16 melody segments are much more representative than others, and implies the result of our method is consistent with the experts' perception and professional knowledge.

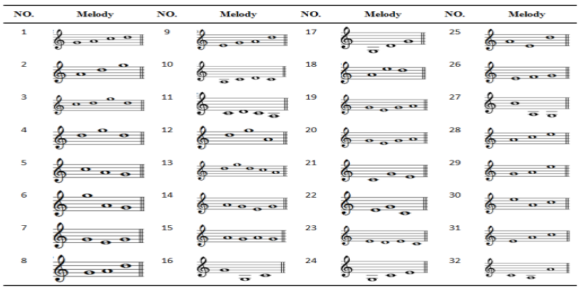


Table 3. Melody segments in the evaluation set.

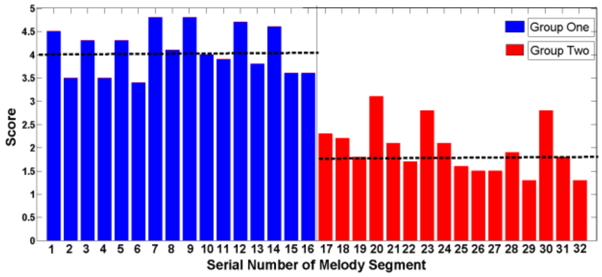


Figure 5. Evaluation results

6. CONCLUSIONS

XinTianYou, an important style of folk songs, has had a profound impact on the musical history of China. The exploration of XinTianYou's general characteristics is important for both automated musical analysis research and obtaining greater insights into Chinese folk music. In this paper, we explore the general characteristics of XinTianYou. Firstly, we built a XinTianYou MIDI database to facilitate future study. Secondly, we propose a new direction of XinTianYou research which examines the music data through its interval combinations (the small melody segments). Thirdly, we propose to use the redundancy-aware top-*k* patterns method to do the clustering. Based on the similarity measure and the support, we finally find the most dominant patterns as the general melodic characteristics.

Each culture has its own evolving heritage. We wonder if these general characteristics are gene of Chinese music. In the future, we can investigate more thoroughly if the general characteristics discovered in this paper are representative across all Chinese folk music.

7. REFERENCES

[1] K. Naoko, et al., “A practical query-by-humming system for a large music database,” in Proceedings of the eighth ACM international conference on Multimedia, 2000, pp.333-342.

[2] Y. Li, Y. Wu, and B. Liu, “A new method for approximate matching melody humming retrieval system and application,” Computer Research and Development, 2004, pp. 1554-1560.

[3] Y. Yang, and X. H, “Cross-cultural music mood classification: a comparison on English and Chinese Songs,” in ISMIR, 2012, pp.19-24.

[4] P.D. León, et al., “Statistical description models for melody analysis and characterization, ” In International Computer Music Conference, 2004, pp.149--156.

[5] C. McKay and F. Ichiro, “Automatic genre classification using large high-level musical feature sets,” in ISMIR, 2004, pp.525-530.

[6] Collins, T., “Improved methods for pattern discovery in music, with applications in automated stylistic composition,” Open University, 2011.

[7] National Editorial Board, Chinese folk songs ·Shaanxi volume, China ISBN Center, 1994.

[8] Z. Liu, “Constraints of traditional music style of the -- Tricolor theory,” Chinese Music, 2014, pp. 59-71.

[9] Y. Wang, “Analysis of the tonal structure in the folk song's melody--take the folk songs of She minority as example (I),” Music Research, 2007(1), pp. 68-74.

[10] Y. Wang, “Analysis of the tonal structure in the folk song's melody--take the folk songs of She minority as example (II),” Music Research, 2007(2), pp. 15-21.

[11] T. Eerola and P. Toiviainen, “MIR in MATLAB: The MIDI Toolbox,” in ISMIR, 2004, pp.22-27.

[12] C. Wu, J. Wang, and C. Chen, “Mining condensed rules for associative classification, ” in Machine Learning and Cybernetics, International Conference on, 2012, pp.1565-1570.

[13] X. Dong, et al., “Extracting redundancy-aware Top-K patterns, ” in Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. 2006, pp.444-453.

[14] Zhao, M., S.J. Turner and W. Cai., “A Data-Driven Crowd Simulation Model Based on Clustering and Classification, ” In Proceedings of the 2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications. 2013, pp.125-134.

[15] S. Zhu, J. Wu, and G. Xia, “Top-K cosine similarity interesting pairs search,” Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on, 2010, pp. 1479 - 1483.

[16] S. Zhu, et al., “Top-MATA: A Max-First traversal method for Top-K cosine similarity search,” in Service Systems and Service Management, 2010 7th International Conference on, 2010, pp.994-998.

[17] A. Madylova and S.G. Oguducu, “A taxonomy based semantic similarity of documents using the cosine measure,” 24th Information Symposium on Computer and Information Sciences, 2009, pp. 129-134.

[18] D. Deng, et al., “Top-K string similarity search with edit-distance constraints,” Data Engineering (ICDE), 2013 IEEE 29th International Conference on, 2013, pp. 925 - 936.

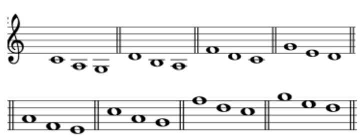
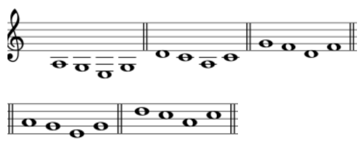
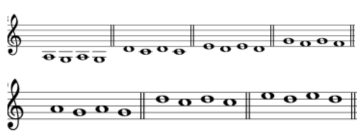
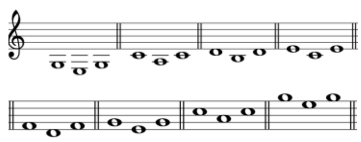
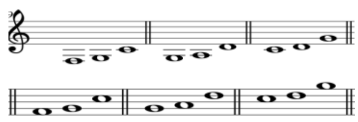
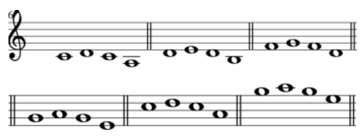
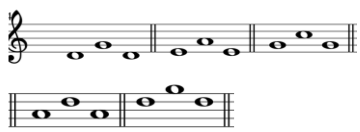
[19] M. Akbar and R.A. Angryk, “Frequent pattern-growth approach for document organization,” in Proceedings of the 2nd international workshop on Ontologies and information systems for the semantic web, 2008, pp.77-82.

[20] N. Gao, et al., “Topic detection based on group average hierarchical clustering,” in Advanced Cloud and Big Data, 2013 International Conference on, 2013, pp.88-92.

[21] S. Gilpin, B. Qian and I. Davidson, “Efficient hierarchical clustering of large high dimensional datasets,” in Proceedings of the 22nd ACM International Conference on information and knowledge management, 2013, pp.1371-1380.

[22] C. Li, Basic Music Theory Teaching Materials, BeiJing: Higher Education Press. 2004

APPENDIX: 16 combinations of intervals and corresponding melodies. First column is the cluster id. Second column is the combination of intervals. Third column is the musical score of corresponding melody segments.

Cluster	**Combination**	**Melodies**
k=1	-3 -2	
k=2	-2 -3 +3	
k=3	-7 -2	
k=4	-2 +2 -2	
k=5	-3 +3	
k=6	-8 +4	
k=7	+2 +3 +2	
k=8	+2 +4	
k=9	+4 +4	
k=10	+3 +2 +4	
k=11	+3 +2 -2	
k=12	+2 +4 -4	
k=13	+2 -2 -3	
k=14	+4 -4	
k=15	+4 -7	
k=16	+4 -4 -2 -3