Classification of Turkish Songs according to Makams by using *n*-grams

Adil Alpkoçak[1] Ali Cenk Gedik[2]

[1] Dokuz Eylul University, Department of Computer Engineering
Buca 35160, Izmir, Turkey
[2] Dokuz Eylul University, Department of Musicology
Narlidere 35340, Izmir, Turkey
{alpkocak@cs.deu.edu.tr, cenk.gedik@emo.org.tr}

Abstract. In Turkish music, *makam* is a concept used to codify phenomena of scale structure, interval structure, and melodic characteristics that underlies composition and improvisation. In this study we propose a 2-level hierarchical classification method using *n*-gram analysis for 10 Turkish *makams; hicaz, mahur, rast, uşşak, buselik, hüseyni, segah, acemaşiran, muhayyer* and *nihavend.* Each makam consisting of 20 recordings, represented as symbolic data. For feature extraction *n*-gram analysis is used to determine discriminative note sequences for each makam. Our main motivation is searching for intrinsic attributes of *makam* which is implicitly defined in theory. Our hierarchical classifier consists of Linear Discriminant Classifier and Decision Trees. As a result we achieved overall success of 98% accuracy for 10 makams.

1 Introduction

Problem of automatic music classification is still being discussed and studied in the community and it seems to be continuing as new approaches emerge.[1] It can be said that there are two main branches in automatic music classification; classification by musical genre and classification by melody. While both are subject to music information retrieval, musical genre classification approaches the problem from the standpoint of the distinct and union properties of musical genres such as classical, jazz, rock, disco etc. on one hand and melody classification studies the similarity properties of melody on the other.

Turkish *makam* music bears an opportunity of having approximately 600 kinds of *makams* in this sense and approximately 30 *makams* are recognizable actually in the community of Turkish *makam* music, both performers and audience. By the way a melody in *makam* music can be classified into one of these 30 *makams* on the contrary to Western music where there is mainly two classification groups such as major and minor. Moreover, this classification study of Turkish music or studies on finding out the intrinsic attributes of *makam* may lead to new solutions towards melodic classification and new approaches to music information retrieval. Both being our motivation, firstly, we concentrated on classification of 10 *makams* which are selected due to their currency, similarity and unsimilarity, in this paper. Features of each *makam* are determined by n-gram analysis, which is primarily used in text retrieval systems. So note sequences are represented by letters and discriminative “words/group of notes” as determined in the case of text retrieval. Linear Discriminant Classifier is used in the first level of hierarchical classification and in the second level with a decision tree.

2 Turkish Makams

In Turkish music, a *makam* (pl. makamlar) is a concept used to codify phenomena of scale structure, interval structure, and melodic characteristics that underlies composition and improvisation [5].

A *makam* can be roughly defined as a scale, as in western music, but additionally the concept of seyir is crucial in the creation of *makam*. This pattern is known in Turkish as *seyir* (meaning basically, "route"). In this study, we considered *seyir* as the melodic movement, where it is the most clearly definition of *seyir* in theory.

Although there are differences among them, various schools classify *makams* into 3 main groups; simple, compound and transposed. Simple *makams* are thought as the main *makam*s that other *makam*s are derived from them, compound *makam*s are composition of two or more *makam*s, and transposed *makam*s which as the name implies are the simple *makam*s transposed to a different tonic.

For this first attempt to automatic classification of Turkish *makam*s, 10 *makam*s are selected due to their popularity, similarity and unsimilarity to achieve a robust classification. It is clear that theoretically similar *makam*s are hard to classify and unsimiliar ones are easier on the other hand. Table 2 demonstrates their relationship in terms of accidentals, tonic, dominant, seyir and their theoretical classification.

Table 1. 10 Turkish *makam*s with their descriptions

***Makam*s**	**Accidentals**		**Tonic**	**Dominant**	**Seyir**	**Roots**
Hicaz	B		A	D	↓-↑	Simple
Acemaşiran	B		F	C	↓	Transposition of Çargah *Makam*
Mahur	F		G	D	↓	Transposition of Çargah *Makam*
Rast	B		G	D	↑	Simple
Uşşak	B		A	D	↑	Simple
Buselik	-	-	A	E	↓-↑	Simple
Huseyni	B		A	E	↓-↑	Simple
Muhayyer	B		A	E	↓	Thought to be Simple and is form of huseyni with falling seyir
Nihavend	-	-	G	D	↓-↑	Transposition of Buselik *Makam*
Segah	B-E		B	D	↑	Composed

From Table 1, it can be seen that *acemaşiran* and *mahur* are similiar by their common roots and *seyir*, the same conditions also hold for *buselik* and *nihavend*. *Muhayyer* and *huseyni* can be similiar as the former is descending *huseyni*. Finally *hicaz, uşşak* and *segah* seem to be unsimiliar. These theoretical considerations will be compared with our automatic classifier's results.

Unlike western music, in Turkish music a whole tone is divided into 9 parts in resulting "comma" pitches. Although MIDI is not capable of representing comma pitches, we solved this problem by converging these notes to nearest ones. So in hicaz *makam* B (4 comma flat) converges to Bb, in mahur F (1 comma sharp) converges to F and 1 comma flats in other *makam*s for B and E converges to B and E respectively.

3 Classifications

In this section, we present the details of our approach to classify Turkish *Makam* music. Our data set is formed of total 200 MIDI files, 20 for each 10 *makam* and classification is applied after necessary preprocessing. So fist, the preprocessing step has be presented before classification process takes place.

3.1. Preprocessing and Representation of Musical Data

Turkish *makam* music can be said to be a monophonic music. In monophonic music only one note is played at a given time. So *makams* can be represented as a string of notes by use of this property, such as "AABBbDE#…" where "b" and "#" are flat and sharp respectively. The whole process of preprocessing is given below:

1. 20 recordings of each *makam* are transposed to tonic pitch C.
2. Each recording is represented in 3 octaves which is a reasonable range to represent *makams*.
3. So each note(range between C3-C6) is represented by a letter from an alphabet set:
   {O,o,R,r,M,P,p,S,s,L,l,b,C,c,D,d,E,F,f,G,g,A,a,B,Q,q,T,t,N,Y,y,Z,z,X,x,h }
4. Repeated notes are discarded due to the investigation of founding unique intervallic sequence for each *makam*. So a recording such as <AaaBffffG> is reduced to < ABfG >. It should be stated that no temporal knowledge is used for representation of *makams*.
5. Finally all recordings become ready to be manipulated as texts for *N*-gram analysis.

Once each song is represented as strings, it becomes reasonable to use N-gram analysis. In text retrieval word can be easily determined by the spaces which brackets the words but in music it is not as easy. Therefore we have searched for the possible all "meaningful" n-grams. As 20 recordings of each *makam* are used, meaningful n-grams correspond to a sequence of n notes which should satisfy the criterion of being included by at least 20 recordings of a *makam*. Therefore these representative sequences are found for each *makam* as features.

Equations from 1 to 4 give the details of the calculation of term weights where document and collection are referred to *makam* and a recording of a *makam* respectively.

f_i^k : term frequency – total number of occurrences of term k in document D_i(eg. a record from hicaz *makam*)

F_k : total number of occurrences of term k in the entire collection (eg. all 20 records of hicaz *makam*)

$$F^k = \sum_{i=1}^{m} f_i^k \quad \textbf{(1)}$$

F_k^d : total number of documents that contain term k for a collection

$$F_k^d = \sum_{i=1}^{m} f_i^d \text{ , where } f_i^d \text{ is } \begin{cases} 1 \text{ if term } k \in D_i \\ 0 \text{ otherwise} \end{cases} \quad \textbf{(2)}$$

"Meaningful" terms which satisfy the criterion $F_k^d = 20$ are selected as mentioned above.

F^c : total number of occurrences of all terms in the entire collection

$$F^c = \sum_{k=1}^{m} F^k \quad \textbf{(3)}$$

w_k : weight of term *k* in the entire collection

$$w_k = F^k / F^c \quad \textbf{(4)}$$

The final feature set is found after detecting *n*-grams for each *makam* which discriminates it from others. Therefore this approach gives rise to a feature set where each feature corresponds to a *makam* and at first hand resulting feature space becomes 10 dimensional. In the next algorithm we explain the details of this feature extraction process.

Steps for N-gram analysis: Document-Matrix

1. For 1<n<5 all *n*-grams are found with total frequency and the number of recordings that each *n*-gram is appeared for each *makam* collection. *n*-gram where n>5 are found to be useless as they have been appeared at most in 15 recordings for a collection.
2. After ranking each *n*-gram for a collection according to the number of recordings it has appeared, highest 10 values are taken into account.
3. 2, 3, 4-grams each now has 10 values showing frequency of occurrence for each *makam* collection.

4. For each *n*-grams all *makams* are evaluated to find the most discriminative ones for each *makam*.
5. Finally for each *n*-gram and *makam* discriminative terms are calculated and these formed the feature set. This last step, determination of feature set can be thought as training phase or more as it is stated in Duda [??]:

"*The conceptual boundary between feature extraction and classification proper is somewhat arbitrary: an ideal feature extractor would yield a representation that makes the job of the classifier trivial; conversely, an omnipotent classifier would not need the help of a sophisticated feature extractor*."

3.2. Classification

Fig.1. shows a brief summary of classification process.

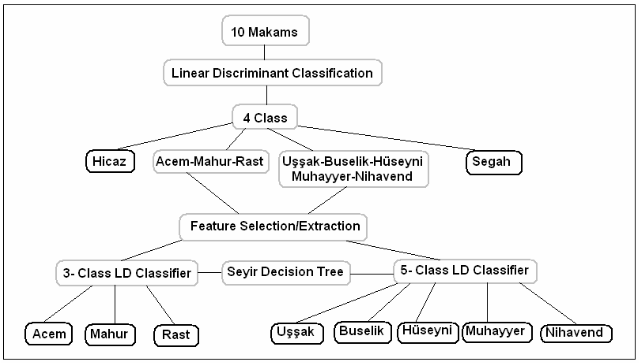

Fig. 1. Block diagram representation of classification process

First Level Classification: Final feature set found from the previous algorithm is shown in Table 2, and we have a 10-dimensional feature space where each feature represents a *makam*. These features are extracted from each labeled recording, that is to say for example frequency of hicaz terms form the value of hicaz feature for a given recording. Number of letters is used to normalize all recordings. One of the interesting results of our classification can be said that it enables to discuss a samples quality as being how much it bears features of other *makam*s. For example, while a sample is classified as hicaz, it can be seen whether it bears some amount of *n*-grams of other *makam* or not.

Linear Discriminant functions gives the class name, the hicaz recording is classified as hicaz in step above. Hicaz and segah are almost true classified and there is a grouping between confused classes.

Acem-mahur-rast forms one group, uşşak-buselik-huseyni-muhayyer and nihavent forms another. This result is in accordance with Turkish *makam* theory as briefly described in Representation chapter. Acem and Mahur are transposed *makam* of Çargah and demonstrates similar *n*-grams with Rast. The same condition is hold for also hüseyni-muhayyer and buselik-nihavend pairs and all have similar *n*-gram results with uşşak.

This level ends up with a conclusion that these groups should be classified in between. So we have four class classification, hicaz, first group *makam*, second group *makam* and *segah*. Our classifiers first level reduces to 4 classes. After making necessary arrangements such as taking mean of feature values for each group, Fig.2. (a) and (b) shows 9 *makam*s in 3-D feature with 1. group, 2. group by segah and hicaz respectively. Again classes are linearly separable.

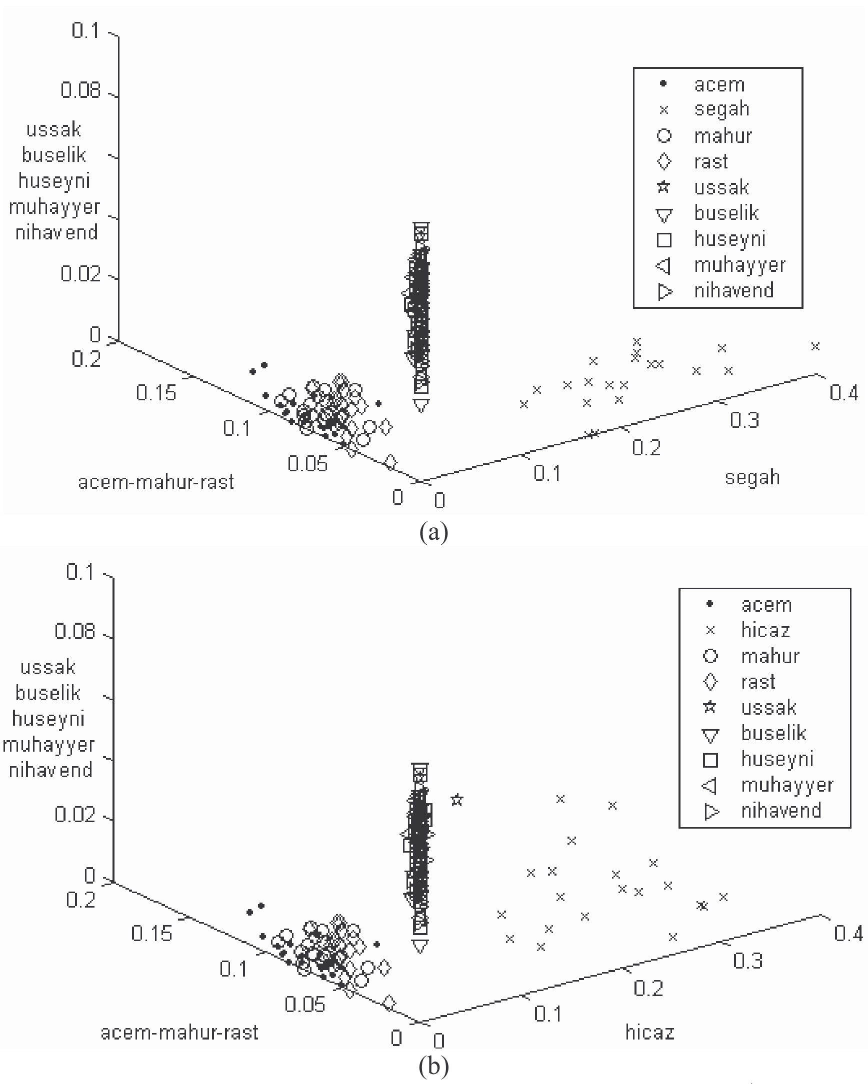

Fig. 2. *Makam* samples in 3-D feature space. Dimensions are (a) segah, 1st group and 2nd group, (b) hicaz, 1st group and 2nd group

Second Level of Hierarchical Classification: Finding discriminative terms for 10 *makam*s was difficult enough but now we have to discriminate acem-mahur-rast in one hand and uşşak-buselik-huseyni-muhayyer-nihavent on the other. Two more classifiers are necessary, 3-class and 5-class classifiers. While this seems to make classification easier in fact to classify these similar *makam*s grouped together naturally needs more than *n*-gram analysis.

N-gram analysis with document matrix is applied again to each group. New document matrixes and naturally new feature sets are found for each group. For classification method used in first level is used together with seyir features of *makam*s. To do this every recordings seyir is obtained by summing the intervallic movements. Ascending movements increases this value and descending movements decrease. Overall sum gives a value representing seyir. Fig.3. shows the graphical representation of seyir.

After applying first level classification method each recording is checked for its seyir value. This can be said to be technique of decision tree. Fig.4. shows the distribution of seyir of ussak and muhayyer so each pair is tried to be discriminated by this additional feature.

LD Classifier and Decision Tree Combination: To clarify this classifier combination it will be necessary and enough to take a look at classification of 1st group, acem, mahur and rast because the process is just the same for the 2. group.

The results of LD classifier are interrogated according to seyir features of samples and *makam*s. After finding the class of a sample from LD classifier, the seyir of sample interrogates this class according to seyir distribution of *makam*s. As the discriminant functions magnitudes are calculated in LD classifier where maximum class function determines the class, this ranking enables the decision making process. In case of contradiction of seyir and assigned class, algorithm assigns the sample to the second maximum functions class.

4 Experimentations

In order to test the performance of our classifier we have selected a total of 200 MIDI files, 20 for each 10 *makams*. The both *makam*s and the songs are selected with a care. and while determining samples from each *makam*, we pay attention to consider various musical forms in *makam* such as *peşrev*, etude, *saz semai* and song to be equally distributed between 10 *makams*. All the codings have been done under MatLab using MIDIToolBox [5].

We evaluated our classifier in terms of sensitivity, selectivity and accuracy for each *makam*. Sensitivity assesses how well the classifier can recognize positive samples. It gives True Positive rate, TP/(TP+FN). Selectivity measures how well the classifier can recognize negative samples and gives True Negative rate TN/(TN+FP), accuracy is a ratio of correctly classified examples to all samples (TP+TN)/(TP+TN+FP+FN). Results can be seen from Table 2.

Table 2. Evaluation results of *makam* Classification.

	True		False		Sensitivity TP/(TP+FN)	Selectivity TN/(TN+FP)	Accuracy (TP+TN)/ (TP+TN+FP+FN)
	(+)	(-)	(+)	(-)			
Hicaz	20	180	0	0	100	100	100
Acem	17	176	3	4	81	98	97
Mahur	16	177	4	3	84	98	97
Rast	20	180	0	0	100	100	100
Ussak	19	180	1	0	100	99	100
Buselik	16	172	4	8	67	98	94
Huseyni	14	177	6	3	82	97	96
Muhayyer	18	176	2	4	82	99	97
Nihavent	15	177	5	3	83	97	96
Segah	20	180	0	0	100	100	100
Total/Avg	**175**	**1775**	**25**	**25**	**87.8**	**98.6**	**97.7**

As a result, our classifiers average performance is %88 for sensitivity, % 99 for selectivity and %98 for accuracy which shows the high success of classification for Turkish *makam*s. This is the final result of our classifier, but if we would evaluate the classifier by its hierarchical levels first level where hicaz and 1st group (acem, mahur, rast) and 2nd group (uşşak, buselik, huseyni, muhayyer, nihavend) and segah are classified to 4-classes has %100 success for all. So confusion arises from the second level of hierarchy where 3-class and 5-class classification takes place within 1st and 2nd group *makam*s.

Finally, we have totally, 21 false classified samples, 13 from transposed relations, 5 from seyir relations and 3 from indirect relations. If we add 4 false classified samples between buselik and hüseyni where we based our indirect explanations number of total false classified samples are found to be 25.

5 Discussions and Conclusion

In this study, we presented an automatic classification of Turkish *makam*s by a hierarchical classifier. In total, 200 MIDI files, 20 samples for each 10 *makam*s, are used and preprocessed to reach a representation which enables mainly a text-retrieval technique *n*-gram analysis for feature selection. It should be mentioned again that we transposed all *makam*s to a unique pitch as if they were western music modes even it is undesired theoretically.

Our study's uniqueness not only comes from being the first classification study of *makam*s but also we thought that the applied classification technique is a new one. This approach is based on using the features being the classes at the same time. Our application of *n*-gram analysis yielded this result and gives the opportunity to consider classes as a degree of membership. Although we left the detailed discussions of this issue in future studies and only stated through in Feature Extraction Section, this degree of membership means that we can consider each sample as bearing different features of different classes of different degrees.

However if our main motivation for *makam* classification is considered we think that the most challenging side of this study would be seen clearly. That is the search for modes to be used in melodic similarity and classification in western music other than major and minor. This study can be said to be the first step of our long term motivation in this meaning. We showed that Turkish *makam*s which are in theory said to be not transposable, can be transposed by converging the comma pitches to their nearest temperament pitches. Although this approach to comma pitches of *makam*s are discussable, we found reasonable to discard them as they all have “1” comma values. By the way it is not necessary to mention that “last words” would be spelled after cognitive tests. At least now we have numerical evidence that *makam*s can be classified without comma pitches. In the second step we are planning to get use of these “modes/*makam*s” for melodic similarity in western music.

References

1. Ozkarahan, E., Database Machines, Prentice-Hall. Inc., 1986.
2. Duda R. O., Hart P. E., Stork D. G., Pattern Classification, Second Edition, Jhon Willey & Sons, Inc., 2001.
3. Gedik, A.C., Alpkocak, A., Işıkhan C., Özer, Y., Türk Makamlarının Otomatik Sınıflandırması, Congress of Representation in Music and Musical Representation, İTÜ, İstanbul.
4. WWW, http://en.wikipedia.org/wiki/*Makam*, visited at March 23, 2006.
5. Eerola, T., Toiviainen, P., MIDI Toolbox: MATLAB Tools for Music Research. University of Jyväskylä: Kopijyvä, Jyväskylä, Finland. Available at http://www.jyu.fi/musica/miditoolbox/, 2004.
6. Aoyagi, T., *Maqam Rast: Intervalling Ordering, Pitch Hierarchy, Performance, and Perception of a Melodic Mode in Arab Music,* PhD Thesis, University of California, Los Angeles, USA, 2001.
7. Gao J., *Linear discriminant model for information retrieval*, Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval SIGIR ‘05, Salvador, Brazil, 2005.
8. Downie J. S., *Evaluating a Simple Approach to Music Information Retrieval: Conceiving Melodic n-grams as Text*, PhD Thesis, University of Western Ontario,USA, 1999.