Automatic Dastgah Recognition Using Markov Models

Luciano Ciamarone¹, Baris Bozkurt², and Xavier Serra²

¹ Independent researcher

² Universitat Pompeu Fabra, Barcelona, Spain

luciano.ciamarone@libero.it, baris.bozkurt@upf.edu, xavier.serra@upf.edu

Abstract. This work focuses on automatic Dastgah recognition of monophonic audio recordings of Iranian music using Markov Models. We present an automatic recognition system that models the sequence of intervals computed from quantized pitch data (estimated from audio) with Markov processes. Classification of an audio file is performed by finding the closest match between the Markov matrix of the file and the (template) matrices computed from the database for each Dastgah. Applying a leave-one-out evaluation strategy on a dataset comprised of 73 files, an accuracy of 0.986 has been observed for one of the four tested distance calculation methods.

Keywords: mode recognition, dastgah recognition, iranian music

1 Introduction

The presented study represents the first attempt in applying Markov Model to a non-western musical mode recognition task. The proposed approach focuses on Persian musical modes which are called *Dastgah*. Several different approaches to the same task have already been documented. In 2011 Abdoli [3] achieved an overall accuracy of 0.85 on a 5 Dastgahs classification task by computing similarity measures between Interval Type 2 Fuzzy Sets. In 2016 Heydarian [5] compared the performances of different methods including chroma features, spectral average, pitch histograms and the use of symbolic data. He reported an accuracy of 0.9 using spectral average and Manhattan metric as a distance calculation method between a signal and a set of templates. Another contribution to the field comes from research works in Turkish music [11] and [4] from which this work inherit some analysis techniques [8]. None of these previous works use Markov models for their classification purposes while for western music several applications has been explored [12][13] although only for chord progression thus under the point of view of music harmony. The presented work, instead, investigates the music melodic aspect developing a mathematical model able to encode the typical melodic interval progression of Persian Dastgahs in the form of Markov Transition Probabilities Matrices (section 3.2). A subsequent distance evaluation method between matrices has been applied in order to carry out the classification task (section 3.3). Finally, standard machine learning evaluation

has been carried to measure system performances. An accuracy of 0.98 has been reached on a database of 73 audio files belonging to the seven main Persian Dastgahs. The complete algorithm has been publicly shared on a github repository[7] for the sake of reproducibility. In the final part of the presented paper future developments of this research have been identified.

2 Persian Traditional Music

Persian music is based on a set of seven principal Dastgahs: *shur*, *homayun*, *segah*, *chahargah*, *mahour*, *rast-panjgah* and *nava*. The seven main modes and their five derivatives (*abu ata*, *bayat-e tork*, *afshari*, *dashti* and *bayat-e esfahan*) are collectively called the twelve dastgahs, they cover most of the structures in Persian music [3] [5].

Traditional Persian music has the octave concept and one octave always contains seven principal notes. The tuning system of Persian music applies 24 quarter tones per octave[2] and does not rely on equal temperament. The most authentic definition has been given by Farhat [1] who teaches that Persian music has very characteristic intervals, one of them is the neutral second. This is a flexible interval but in all its variations it is noticeably larger than the minor second (western semitone) and smaller than the major second (western whole tone). Another typical interval [1] is an interval which is larger than the major second (western whole tone) but smaller than the western minor third. Rhythmic structure of Persian music is generally strictly connected to voice and speech, often music is conceived as accompaniment for singers. Each *Dastgah* consists of some partial melodies called *Gushe*, the arrangement of *Gushes* during the performance is called *Radif*. Conceptually Persian music is conceived like melodic motives around a central tone and modulation is conceived as changing the central tone.

3 Methodology

The presented Dastgah recognition system applies a sequence of three processes: pitch estimation and quantization, Markov Modeling, and classification. Markov chains have been used for modeling the sequence of musical intervals. The concept behind Persian musical intervals is the same as in western music, which is a frequency ratio between two notes. For example the western musical fifth is defined as $\sqrt[12]{2^7} \cong 3/2$.² The presented Markov algorithm models in the form of Transition Probability Matrix, the sequence (in time) of musical intervals contained in each audio file. The synthetic example in Fig. 1 can help understanding the Markov Matrix building strategy. Frequency values in time like in Fig. 2 are converted in a vector of consecutive musical intervals; in this example $[3/2, 3/2, 2, 2, 2, 3, 3, 2]$. Thus we have three Markov states: $A=3/2$, $B=2$ and $C=3$.

² an example of western fifth is the interval between A4=440Hz and E5=659.25Hz ($659.25/440=1.498 \cong \sqrt[12]{2^7}$)

Finally, transitions between consecutive states are counted and cumulated to build up the Markov matrix. As we can see transitions from B to A never occurs (neither C to A and A to C), furthermore if we had a state $D=5/2$ it would have occurrence count equal to 0 in this example.

Fig. 1. Example of Markov Matrix building strategy starting from quantized pitch data (this is not real data)

3.1 Pitch Detection and Quantization

For pitch detection, we have used the algorithm explained in [10] (more specifically the implementation by its authors³) which mainly applies post-filters to the output of the Melodia algorithm [9]. According to the implementation³ of [10], pitch quantization is achieved by mapping each pitch value to the closest pitch histogram peak where pitch histograms are computed as in [8]. Frequency quantization has the effect of stabilizing the values given by the pitch detection algorithm forcing them to be equal to the closest pitch histogram peak [8] (for example the sequence of pitches: [200.1, 200.2, 199.85] is forced to be [200, 200, 200] if 200Hz is the closest pitch histogram peak).

Secondly, spurious and very short duration notes has been removed because they are assumed to be detection errors. When a quantized note is removed (because of too short in duration) it is replaced (and merged) with the following stable note (a stable note is assumed to be at least 15 milliseconds long). The quantized pitch data obtained as a result of these two post processing steps are exemplified in Fig. 2 .

3.2 Markov Model

The Markov Model builder block uses the data generated by the pitch detection and quantization algorithm in order to create a vector of musical intervals which are calculated from the ratio between two consecutive frequency values (as explained at the beginning of this *section 3* and in Fig. 1). The resulting sequence of musical intervals (from one audio file) can be concatenated (using

³ <https://github.com/MTG/predominantmelodymakam>

Fig. 2. Ten seconds of a Segah audio sample. *Dashed Line:* Melody profile of the audio input as analyzed by the Essentia monophonic pitch detection algorithm. *Continuous Line:* Same melody profile after post processing for stabilizing the frequency values. *Dotted Circles:* Peaks that generated too short duration quantized notes which have been replaced and merged with the following stable note.

several audio files) in case one wants to build up the database matrix associated with one of the seven possible Dastgah considered in the present work.

Building the Transition Probability Matrix associated with the sequence of musical intervals consists in counting the number of transition occurrences between consecutive intervals (as exemplified in Fig. 1). All probability values have been normalized to one in the end. After that a vector containing the count of the occurrences of each Markov Model state has been created, the Fig. 3 shows this vector calculated for a small audio chunk belonging to Dastgah-e Segah.

The vector plotted in Fig. 3 is a sort of one-dimensional view of a Markov Matrix, on the x axis there are the Markov Matrix indexes from one to ns (where ns is the total number of Markov States); on y axis there is the normalized frequency of occurrence of each state. The presented Dastgah Recognition system does not use a previously defined TET musical system and there are no octave folding operations. The dimension of the Markov Matrix just needs to be greater than the maximum number of playable musical intervals and sufficiently small not to overload the computational cost of the whole algorithm. Results presented in this paper have been obtained using 98 Markov States ($ns=98$). Each Markov State represents a bin whose width progresses exponentially with a base of $c = (\frac{1}{ns} + 1)$ ⁴, the computed musical intervals are mapped into these bins (this

⁴ using 98 Markov States, $c = (\frac{1}{98} + 1) = 1.0102$ which is smaller than the western music semitone $\sqrt[12]{2} = 1.0595$

Fig. 3. Markov states frequency of occurrence for a small chunk of Dastgah-e Segah

means that small fluctuations of frequency ratios are mapped into the same bin if those fluctuations are contained in the bin width)[7]. In *Fig. 3* there are lots of Markov states with number of occurrences equal to zero, this means that for that audio sample, no musical intervals have been mapped into those bins, this is why this vector can be considered as a first raw fingerprint of the Dastgah to which the audio file belongs to.

3.3 Classification

In order to classify one unknown audio file as belonging to one of the seven considered families of Dastgah it is necessary to implement a metric able to measure the distance between the matrix associated to the unknown audio file and the seven database matrices associated to the seven Dastgah. In this work four distance candidates have been tested: Euclidean distance (*equation 1*), Kullback-Leibler distance (*equation 2*), Bhattacharyya likelihood (*equation 3*) and the last metric (which is also in the form of a likelihood) State Probability Correlation (*equation 4*). This last metric basically performs the dot product between *Frequency of Occurrence* vectors like the one showed in *Fig. 3*.

$$euclidean = \sqrt{\sum_{i=1}^{ns} \sum_{j=1}^{ns} (\chi_{ij} - db_{ij})^2}. \quad (1)$$

$$kullback = \sum_{i=1}^{ns} \sum_{j=1}^{ns} \chi_{ij} \cdot \log \left(\frac{\chi_{ij}}{db_{ij}} \right). \quad (2)$$

$$battacharyya = \sum_{i=1}^{ns} \sum_{j=1}^{ns} \sqrt{\chi_{ij} \cdot db_{ij}}. \quad (3)$$

$$SPC = \sum_{n=1}^{ns} \bar{\chi}_n \cdot \bar{db}_n . \quad (4)$$

Where ns is the number of Markov states, χ_{ij} is one element (scalar) of the unknown matrix, db_{ij} is one element (scalar) of the database matrix; $\bar{\chi}_n$ and $\bar{db}_n$ are the cumulated values (scalars) along the rows of the matrices in order to obtain a cumulative value of occurrence probability for each Markov state.

4 Experiments and Results

The experiments carried out had the goal of testing and validating the classification algorithm, a standard machine learning evaluation procedure has been applied. A *Leave One Out* testing strategy has been implemented and in the end, standard machine learning evaluation parameters have been calculated. A github repository [7] has been created where the testing package can be downloaded and executed again for obtaining the same results presented in this paper.

4.1 Dataset

In the aforementioned github repository an *annotation.json* file can be found. In this file is contained the formal description of the used database of audio files. A total amount of 73 audio files has been collected with average duration of about two minutes each (maximum 4:50, minimum 1:10), they belong to the seven Dastgah category considered in this work: Segah (16 files), RastPanjgah (5 files), Nava (7 files), Mahour (15 files), Hodayun (10 files), Chahargah (8 files) and Shur (12 files). The recordings are monophonic (recording of a single instrument) and does not include mode transitions/modulations. No constraint has been applied in selection of instruments. The dataset contains recordings of singing, tar, setar, santur among other instruments.

4.2 Testing Strategy and Results

A *Leave One Out* validation procedure has been considered for the presented work. The procedure uses one sample for testing and all the rest for modeling. This is repeated for each sample and the results are averaged. At the end of validation process the following standard machine learning evaluation measures has been calculated: Recall, Specificity, Precision, False Positive Rate, False Negative Rate, F1 and Accuracy. The four different distance calculation methods are defined in equations 1, 2, 3 and 4. Table 1 shows the evaluation measures obtained for each distance metric.

The first two methods gave very poor results below the 50% of correct answers. The logarithmic terms in *equation 2* makes this metric particularly suitable for very smooth distributions like Gaussian ones; in our case this metric suffers the big entropy (from a statistical point of view) of the data it is applied to. The last method (SPC) gave an *Accuracy* of 0.507 and a *FPR* (*False*

Fig. 4. Confusion matrix for the Bhattacharyya likelihood calculation method

Positive Rate) of 0.493 . The third distance calculation method (*Bhattacharyya likelihood*) gave instead an *Accuracy* of 0.986 and a *FPR* of 0.014 . Fig. 4 shows the confusion matrix for the Bhattacharyya likelihood.

Results clearly state that *Equation 3* is the best way of calculating similarities between Markov Transition Probability Matrices; in fact this method resulted in only one error on 73 audio file which means a percentage of correct answers equal to 98.6%. The only error of the *Bhattacharyya* classification method is an audio file belonging to the *Mahour* family which has been classified as *Nava*. The reason why all the other metrics are so far from the *Bhattacharyya* metric in terms of results, is that *equation 3* is the only one which uses an *element by element* product between Markov Matrices, this produces the effect that only values different from zero contribute to the total sum; non-zero values represent specific transitions between consecutive musical intervals and these transitions are the musical core of *Dastgahs*.

	metric	recall	specificity	precision	FPR	FNR	F1	accuracy
	euclidean	0.061	0.663	0.247	0.753	0.939	0.098	0.247
	kullback	0.014	0.306	0.068	0.931	0.986	0.024	0.068
	battacharyya	0.935	0.998	0.986	0.014	0.065	0.960	0.986
	SPC	0.170	0.860	0.507	0.493	0.829	0.255	0.507

Table 1. Scores

5 Conclusions and Further Developments

In this work we approached the problem of Persian Dastgah recognition and classification using Markov Models. The presented Dastgah recognition system has been tested following a standard machine learning evaluation procedure and it gave a maximum accuracy of 0.986. Results show that Markov Models are able to encode information about the content of each Dastgah in terms of musical intervals and their temporal sequence. The presented system has been tested on monophonic recordings of short duration. Our further efforts will be dedicated to building a larger dataset including longer recordings and improvisations as well as multi instrumental recordings.

References

1. Farhat, H., The dastgh concept in Persian music, Cambridge University Press, Cambridge, 1990.
2. Vaziri, A. N.: *Dastur-e Tar*, Tehran, 1913.
3. Abdoli, S., Iranian traditional music dastgh classification, Proceedings of the 12 th International Conference on Music Information Retrieval (ISMIR), Miami, 2011.
4. B. Bozkurt, R. Ayangil, A. Holzapfel, 2014, "Computational Analysis of Turkish Makam Music: Review of State-of-the-Art and Challenges, Journal of New Music Research, 43:1, pp. 3-23.
5. Heydarian, P., Automatic recognition of Persian musical modes in audio musical signals, PhD thesis, London Metropolitan University, 2016.
6. C++ library for audio and music analysis, description and synthesis, including Python bindings, [<https://github.com/MTG/essentia>].
7. Dastgah-Recognition-System github repository, [<https://github.com/luciamarock/Dastgah-Recognition-System>].
8. A. C. Gedik and B. Bozkurt, Pitch-frequency histogram-based music information retrieval for Turkish music, Signal Processing, vol. 90, no. 4, pp. 10491063, 2010.
9. J. Salamon and E. Gomez, Melody extraction from polyphonic music signals using pitch contour characteristics, in IEEE Transactions on Audio, Speech and Language Processing, vol. 20, no. 6, 2012, pp. 17591770.
10. H. S. Atli, B. Uyar, S. Senturk, B. Bozkurt, and X. Serra, Audio feature extraction for exploring Turkish makam music, in Proc. of 3rd Int. Conf. on Audio Technologies for Music and Media, Ankara, 2014, pp. 142153.
11. S. Senturk, Computational analysis of audio recordings and music scores for the description and discovery of Ottoman-Turkish makam music, Ph.D. dissertation, Universitat Pompeu Fabra, 2016.
12. Noland, K., Computational tonality estimation: signal processing and Hidden Markov Models, PhD thesis, Queen Mary, University of London, 2009.
13. Noland, K. and Sandler, M., Key estimation using a Hidden Markov Model. Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR), Victoria, Canada, 2006.