MORTY: A Toolbox for Mode Recognition and Tonic Identification

Altuğ Karakurt
The Ohio State University
Columbus, OH 43210
karakurt.1@osu.edu

Sertan Şentürk
Universitat Pompeu Fabra
Roc Boronat, 138
Barcelona, Spain 08018
sertan.senturk@upf.edu

Xavier Serra
Universitat Pompeu Fabra
Roc Boronat, 138
Barcelona, Spain 08018
xavier.serra@upf.edu

ABSTRACT

In the general sense, mode defines the melodic framework and tonic acts as the reference tuning pitch for the melody in the performances of many music cultures. The mode and tonic information of the audio recordings is essential for many music information retrieval tasks such as automatic transcription, tuning analysis and music similarity. In this paper we present MORTY, an open source toolbox for mode recognition and tonic identification. The toolbox implements generalized variants of two state-of-the-art methods based on pitch distribution analysis. The algorithms are designed in a generic manner such that they can be easily optimized according to the culture-specific aspects of the studied music tradition. We test the generalized methodology systematically on the largest mode recognition dataset curated for Ottoman-Turkish makam music so far, which is composed of 1000 recordings in 50 modes. We obtained 95.8%, 71.8% and 63.6% accuracy in tonic identification, mode recognition and joint mode and tonic estimation tasks, respectively. We additionally present recent experiments on Carnatic and Hindustani music in comparison with several methodologies recently proposed for raga/raag recognition. We prioritized the reproducibility of our work and provide all of our data, code and results publicly. Hence we hope that our toolbox would be used as a benchmark for future methodologies proposed for mode recognition and tonic identification, especially for music traditions in which these computational tasks have not been addressed yet.

Keywords

Mode recognition; Tonic Identification; Toolbox; Ottoman-Turkish makam music; Carnatic Music; Hindustani Music; Pitch Class Distribution; k-nearest neighbors classification; Open Source Software; Reproducibility

1. INTRODUCTION

In many music cultures, the melodies adhere to a particular melodic framework, which specifies the melodic characteristics of the music. While the function and the understanding of these frameworks are distinct from a culture-specific perspective, in a broader sense they may be considered as the “modes” of the studied music culture. Some of the music traditions that can be considered as “modal” are Indian art musics, the makam traditions and medieval church chants [20]. Mode recognition is an important complementary task in computational musicology, music discovery, music similarity and recommendation.

Tonic is another important musical concept. It acts as the reference frequency for the melodic progression in a performance. In many music cultures there is no standard reference tuning frequency, which makes it crucial to identify the tonic frequency to study melodic interactions. Estimating the tonic of a recording is the first step for various computational tasks such as tuning analysis [7], automatic transcription [4] and melodic motif discovery [16].

There has been a extensive interest on mode recognition in the last decade [17]. Most of these work focus on culture-specific approaches for music traditions like Ottoman-Turkish Makam music (OTMM) [13], Carnatic music [11, 12, 16], Hindustani music [9, 10, 15] and Dastgah music [1]. A considerable portion of these studies are based on comparing pitch distributions [9, 10, 11, 12, 13], which are shown to be reliable in the mode recognition task. There also exists recent approaches that are based on characteristic melodic motif mining using network analysis [15, 16], aggregating note models using automatic transcription [18] or audio-score alignment [23] and classification using neural networks [24, 26], all of which are designed specific to the studied music culture and are not generalizable to other music cultures without considerable effort. Similarly, several studies on tonic identification use pitch distribution based methods [6, 10]. More recently there has been an interest in culture specific methods for this task [2, 14, 22] that make use of heuristics and the musical characteristics of the studied tradition.

In these studies, the features extracted from the data,[1] source code and the experimental results are not usually shared. We consider the unavailability of public tools, datasets and reproducible experimentation as major obstacles for computational music information research, especially if the relevant tasks have not been applied to studied music traditions earlier.

We present MORTY (**MO**de **R**ecognition and **T**onic **Y**dentification Toolbox), an open source toolbox written in *Python*

3rd International Digital Libraries for Musicology workshop (DLfM 2016) August 12 2016, New York, USA

© 2016 ACM. ISBN 978-1-4503-4751-8/16/08…$15.00

DOI: http://dx.doi.org/10.1145/2970044.2970054

[1] Excluding the commercial audio recordings, which cannot be generally made public due to copyright laws.

for mode recognition and tonic identification. It contains a generalized implementation of two pitch distribution based methods proposed for Ottoman-Turkish makam music [6, 13] and Hindustani music [10]. Our primary aim is to provide open and flexible tools for the mode recognition and tonic identification tasks, which can be applied to different music cultures while allowing the users to optimize the parameters easily according to the characteristics of the studied music. MORTY may be (and has been, see Section 6) used as a benchmark against novel methodologies proposed.

Another motivation for this work is to provide tools for several relevant tasks such as tuning and intonation analysis [13]. Combined with automatic tonic identification and makam recognition, these features would facilitate the curation and description of large audio corpora not only by greatly reducing the time and effort spent on manual annotations but also providing automatically extracted, reliable information, which would be too difficult or time-consuming for human annotators.

Our contributions can be summarized as:

1. An open toolbox aimed to set a benchmark for future research in mode recognition and tonic identification, which implements and generalizes the state of the art methodologies proposed by Bozkurt and Gedik [6, 13], and Chordia and Şentürk [10]
2. The largest makam recognition dataset for Ottoman-Turkish Makam Music (OTMM), composed of 1000 audio recordings from 20 makams with annotated tonic frequency and editorial metadata.
3. Exhaustive and reproducible evaluation of the aforementioned two state of the art methods on the Ottoman-Turkish makam recognition dataset.
4. Improving the state of the art in tonic identification method applied to OTMM
5. Mode recognition experiments on Hindustani and Carnatic music traditions to demonstrate the applicability of our implementations on different music cultures.

The rest of this paper is organized as the following: Section 2 provides formal definitions of the problems. Section 3 presents the implementation details and the features of our toolbox. Section 4 describes the two state of the art methods and their implementations in detail. Section 5 explains the experiments and the obtained results on our OTMM dataset and Section 6 presents our results on Carnatic and Hindustani cultures. Finally, we discuss the results we obtained in Section 7 and conclude with our comments and suggestions for the future work in Section 8.

For the sake of open research and reproducibility, the toolbox (Section 3) the dataset (Section 5.1), experiments (Section 5) and results (Section 5.3) are accessible publicly via CompMusic website.[2]

2. PROBLEM DEFINITION

We define *mode recognition* as classifying the mode $\zeta^{(a)}$ of an audio excerpt (a) from a discrete set of modes $Z := \{\zeta_1, \ldots, \zeta_V\}$, where $\zeta^{(a)} \in Z$ and V is the total number of modes. The mode set is specific to music culture being studied. In mode recognition, we assume that the tonic frequency $r^{(a)}$ of the audio recording is available.

We define *tonic identification* as estimating the frequency or the pitch class (if the octave information of the tonic is not well-defined for the music culture or the performance) of the performance tonic. We denote the tonic of an audio excerpt as $r^{(a)}$. Unlike mode, tonic is a continuous variable. However, in practice, the tonic is typically constrained to be one of the stable pitches or pitch classes performed in the audio excerpt [10, 13]. With this assumption, tonic identification can be reformulated as estimating the tonic frequency or the pitch class $r^{(a)}$ from a finite set of stable frequencies/pitch-classes $R := \{r_1, \ldots, r_W^{(a)}\}$ performed in an audio excerpt (a), where $r^{(a)} \in R$ and $W^{(a)}$ is the number of the stable pitches in the audio excerpt. In tonic identification, we assume that the mode $\zeta^{(a)}$ of the recording is known.

A third scenario arises when both the tonic $r^{(a)}$ and the mode $\zeta^{(a)}$ of the recording (a) are unknown. In this case, we identify the tonic and recognize the mode together, which we term as *joint estimation*.

Note that these scenarios are actually multi-class problems, since the mode and the tonic may change throughout the performance. This is a more challenging problem, where we would not only like to obtain the set of the modes and tonics in the performance but also mark the intervals, where these musical “attributes” are observed.[3] There has not been any generalizable method proposed for either mode recognition or tonic identification in such a scenario yet. In MORTY, we restrict the problem on mode recognition and tonic identification of audio excerpts with a single mode and tonic, and leave the multiple estimation problem as a future work to investigate.

3. MORTY

MORTY (**MO**de **R**ecognition and **T**onic **Y**dentification Toolbox) is free software, licensed under *Affero GPLv3*.[4] It is implemented in *Python* 2.7 and uses the open source *NumPy* and *SciPy* libraries for numeric computations, *scikit-learn* for machine learning related tasks and *Essentia* [5] for audio processing. Since our motivation is handling large audio collections like digital libraries, we also provide parallelization through *ipyparallel*, a part of *Jupyter* project[5].

As a user manual, we provide *Jupyter* notebooks that demonstrate example usage for each method (Section 4), as well as examples for parallelization.[6] The toolbox was implemented with a modular approach such that it is easy to modify and extend, which makes it possible for future users to contribute with new features, as well as to customize the implementations according to their needs.

4. METHODOLOGY

In MORTY we combine and generalize the two state of the art methods, originally proposed for audio recordings of OTMM [6, 13] and short audio excerpts of Hindustani music [10]. The generalized methods are supervised and use k-nearest neighbors (k-NN) estimation for classification.

[2] http://compmusic.upf.edu/node/319

[3] A manually annotated example for OTMM is given in http://musicbrainz.org/recording/37dd6a6a-4c19-4a86-886a-882840d59518

[4] https://github.com/altugkarakurt/morty

[5] https://jupyter.org/

[6] https://github.com/altugkarakurt/morty/blob/e927386dc72e6282f0ccfaf1c390625f2a554268/demos/knn_demo.ipynb

Our implementations are generic such that the parameters selected in the feature extraction, training and testing steps can be optimized for the properties of the studied music tradition. We also allow the user to classify either short audio excerpts or complete audio recordings and switch between different features, training schemes and tasks as introduced in [6, 10, 13].

In the training step we use audio excerpts with annotated mode and tonic. We first extract a predominant melody for each audio excerpt. These are used to compute either pitch distributions (PD) or pitch class distributions (PCD) (Section 4.1). Next, we create mode models from these computed distributions (Section 4.2).

Given an audio recording with an unknown mode and/or tonic, we extract its predominant melody and compute the distribution. Then, we compute a distance or dissimilarity between the distribution of the test audio and the selected distributions in the training models and compute the k nearest neighbors according to the computed measure (Section 4.3). Finally, we estimate the unknown mode and/or tonic as the most common candidate among the k nearest neighbors (Section 4.4-4.6).

Now we proceed to explain the generalized methodology in detail. We label the tasks, features, training models and parameters in MORTY explicitly with the letters **T, F, M** and **P** throughout this Section for the sake of clarity.

4.1 Feature Extraction

The first step of method is predominant melody extraction **(F1)** [6, 10, 13]. As discussed in [3] and [6], the quality of the extracted pitch predominant melody directly affects the reliability of the computed models and predominant melody extraction methods optimized or designed for the culture-specific aspects of the studied music might be desirable at this step. The implementation of such an algorithm is outside the scope of MORTY.

We denote the predominant melody extracted from an audio excerpt, (a), as $X^{(a)} \triangleq \left(x_1^{(a)} \dots x_I^{(a)}\right)$, where $x_i^{(a)} \in X^{(a)}$ is a pitch sample and $i \in [1 : I]$, where I is the length of the predominant melody.

Next, the samples in the predominant melody are converted to the cent scale using the equation below:

$$x^{(r)} \triangleq 1200 \log_2 \left(\frac{x}{r}\right) \tag{1}$$

Here, $x^{(r)}$ denotes the cent distance of the frequency x from the reference frequency r. In the training step (Section 4.2) and the mode recognition task (Section 4.4), i.e. when the annotated tonic is available, r is the annotated tonic frequency of the audio excerpt. In the tonic identification and joint identification tasks the predominant melody will be normalized with respect to the several tonic candidates one by one, one of which will be identified as the tonic (Section 4.5).

Using the normalized predominant samples we compute either a pitch distribution (PD) as used in [13] or a pitch class distribution (PCD) as used in [10]. PD and PCD shows the relative occurrence of the pitch and pitch class values with respect to each other, respectively. Throughout the text we simply refer the PDs and PCDs collectively as "distributions" **(F2)**. The values in both distributions are computed as:

$$h_n \triangleq \frac{\sum_{i=1}^{I} \lambda_n (x_i)}{I} \tag{2}$$

where h_n is the occurrence computed for the n-th bin in the distribution h, computed samples $x_i \in X$ in the normalized pitch and I is the number of pitch samples.

The accumulator function λ for PDs is defined as:

$$\lambda_n(x) \triangleq \begin{cases} 1, & c_n \leq x \leq c_{n+1} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where x is a normalized pitch sample and (c_n, c_{n+1}) are the boundaries of the n-th bin. Similarly the λ function for PCDs is defined as:

$$\lambda_m(x) \triangleq \begin{cases} 1, & c_n \leq (x \bmod 1200) \leq c_{n+1} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Note that the PCD is a "circular" feature, e.g. the first and the last bins are next to each other. Also notice that both PD and PCD are normalized such that the resultant distribution can be treated as a probability density function.

The bin size β **(P1)** of the distribution determines how precise the distribution is (to the extend allowed by the cent-precision of the predominant melody) in representing the pitch space, the tuning of the stable pitches and the microtonal characteristics in a lower-level. The computed distributions might need to have a small bin size, e.g. less than a quarter tone (50 cents) for many music cultures [10, 13]. We select a constant bin size for the computed distributions, i.e. $\beta = c_{n+1} - c_n, \forall n$. The bin centers of both PDs and PCDs are selected such that the reference frequency r is represented as a bin centered around 0 cents. We denote the number of bins in a distribution as N. Note that N equals to $\lfloor 1200/\beta \rfloor$ in a PCD.

To remove the spurious peaks in the distribution, we convolve it with a Gaussian kernel and obtain a "smoothed" distribution [10]. The standard deviation of the Gaussian kernel, termed as the kernel width σ **(P2)**, determines how smooth the resulting distribution will get. The kernel width should be comparable to the bin size **(P1)** since a value lower than one third of the bin size would not contribute much to smoothing[7] and a high value would "blur" the distribution too much. Moreover, this parameter has a direct impact on the number and the location of tonic candidates in tonic identification (Section 4.5), which might effect both the accuracy and the processing time. In our implementation, we select the overall width of the Gaussian kernel as 5 times the kernel width from peak to tail for performance reasons.

4.2 Training Model

As mentioned earlier, the implemented method is supervised and hence require training data, i.e. audio excerpts with the annotated mode and tonic. From a training audio excerpt (a), we first extract the predominant melody $X^{(a)}$ and normalize with respect to the annotated tonic frequency $r^{(a)}$ (Equation 1). Next, the normalized predominant melodies $X^{\left(a, r^{(a)}\right)}$ are grouped according to the annotated mode $\zeta^{(a)}$ of each individual excerpt.

[7] The values of the bins in a Gaussian kernel, which are more than three standard deviations away from the mean are greatly diminished.

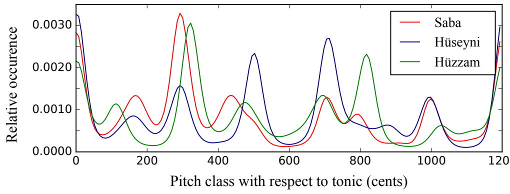

Figure 1: An example model with single PCD per mode trained for three OTMM makams

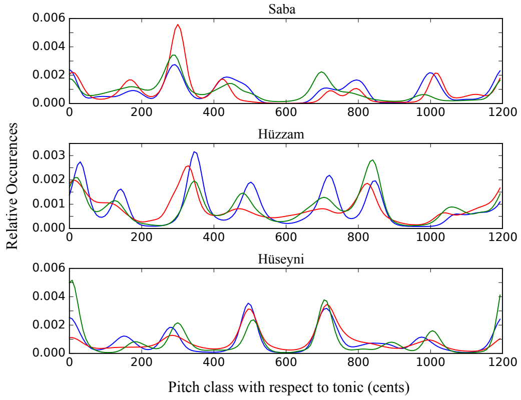

Figure 2: An example model with three PCDs per mode trained for three OTMM makams

The fundamental difference between the methods proposed in [10] and [13] is the training model **(M)** obtained in the training step. The methodology proposed in [6, 13] joins all the normalized predominant melodies and compute a single distribution per mode. On the other hand, [10] creates a separate distribution from each annotated excerpt (a). From a machine learning perspective [13] represents each mode with a single data point (Figure 1), whereas [10] represents them with many (Figure 2) in an N-dimensional space, where N is the number of bins in the distributions. From now on, we term the training models using the training step in [6, 13] and [10] as "single distribution per mode" and "multi-distributions per mode", respectively. We denote the obtained model as $M \triangleq \{m_1, m_2, \ldots\}$. $m_j \in M$ is a tuple $\langle h_j, \zeta_j \rangle$, where h_j and ζ_j denotes the trained distribution and the mode label of m_j, respectively. The model M consists of the distribution representations for V modes, where V is the number of unique mode labels $\zeta_v, v \in \{1,..,V\}$ in the training excerpts.

4.3 Nearest Neighbor Selection

In mode recognition, tonic identification and joint estimation tasks (Section 4.4-4.6), the common step is to find the nearest neighbor(s) of a selected distribution among a set of distributions to be compared against. To find the nearest neighbors we compute a distance or a dissimilarity between the test distribution and each distribution in the comparison set [8]. We have currently implemented the distance and the similarity metrics in [13, 10], namely, City-Block (L_1 Norm) distance, Euclidean (L_2 Norm) distance, L_3 Norm, Bhattacharyya distance, intersection and cross correlation **(P3)**. Note that intersection and cross correlation are similarity metrics, hence we convert them to dissimilarities (i.e. $1-$similarity) instead. The choice of the distance or dissimilarity measure plays a crucial role in the neighbor selection.

After the distances or the dissimilarities are computed, the compared distributions are ranked and the k **(P4)** nearest neighbors are selected. We then estimate the test sample as the most common label of the neighbors. In case of a tie between two or more groups, we select label of the group, which accumulates the lowest distance or dissimilarity. Note that if a single-distribution is computed for each mode (**M** as explained in Section 4.2), the k value is always 1, since each mode is only represented by one sample.

Now we proceed to explain the procedure for each task **(T)** in detail.

4.4 Mode Recognition

Given an audio excerpt (b) with an unknown mode, we compute the distribution $h^{(b,r^{(b)})}$ by taking the annotated tonic $r^{(b)}$ as the reference (Section 4.1). Next we compute the distance or the dissimilarity between $h^{(b,r^{(b)})}$ and the trained distribution h_j of each m_j, $\forall m_j \in M$, where M is the trained model, and obtain the k nearest neighbors to (b). We estimate the mode of (b) as the most common label ζ_v within the nearest neighbors as explained in Section 4.3.

4.5 Tonic Identification

Given an audio excerpt (b) with the annotated mode $\zeta^{(b)}$, we first extract the predominant melody X^b. Then we compute a distribution $h^{(b,*)}$ by taking an arbitrary frequency $(*)$ as the reference frequency (Section 4.1). We detect the peaks in the distribution using the method explained in [25]. The peaks indicate the stable pitches performed in the excerpt. We only consider the peaks, which have a ratio between its height and the maxima of the distribution above a constant threshold **(P5)**. We denote the set of tonic candidates as $R \triangleq \{r_1, \ldots, r_{W^{(b)}}\}$, where $W^{(b)}$ is the number of detected peaks. The cent distance between r_w and $*$ (Equation 1) is given as $r_w^{(*)} := (n_w - n_*) \times \beta$, $\forall l \in W^{(b)}$, where β denotes the bin size **(P1)** of the distribution **(F2)**, and n_w and n_* are the bin indices, in which l and $*$ reside in, i.e. $\lambda_{n_l}(l) = \lambda_{n_*}(*) = 1$ (Equation 2). Assuming each of the peaks r_w as the tonic candidate, we shift the distribution $h^{(b,*)}$ and obtain $h^{(b,r_w)}$ such that the n-th bin becomes the $(n+n_*-n_w)$-th for the PDs and the $(n+n_*-n_w \, mod \, N)$-th (where N is the total number of bins) for the PCDs, respectively and the n_w-th bin represents 0 cents in the shifted distribution.

From the training model M, we select all the $m_j \in M$ with the label $\zeta^{(b)}$. Next we compute the distance or the dissimilarity between each shifted distribution $h^{(b,r_w)}$ and the selected m_js. We select the k pairs with the lowest distance or dissimilarity and select the most occurring peak r_w in the neighbors as the estimated tonic (Section 4.3).

4.6 Joint Estimation

Given an audio excerpt (b) with unknown mode and tonic, we compute the tonic candidates, $R \triangleq \{r_1, \ldots, r_{W^{(b)}}\}$ and the distributions $h^{(b,r_w)}$ assuming each $r_w \in R$ as the tonic candidate as explained in Section 4.5. Next we compute the distance or the dissimilarity between each pair of shifted distribution $h^{(b,r_w)}$ and $m_j \in M$. We select the k pairs with the lowest distance or dissimilarity and estimate the most occurring $\langle$mode, tonic candidate$\rangle$ pair, i.e. $\langle \zeta_v, r_w \rangle$ as the mode and the tonic (Section 4.3).

5. EXPERIMENTS ON OTMM

In this section, we provide the results of the experiments we did with the implementations provided in MORTY. We did exhaustive experiments using our dataset to demonstrate some properties of these methods, find the best parameter sets for OTMM and to provide some heuristics for future users. For the sake of reproducibility all of the scripts, computed features, experiments and results are shared online.[8]

5.1 Test Dataset

In [13], the makam recognition methodology was evaluated on 172 solo audio recordings in 9 makams. To the best of our knowledge, this dataset represents the biggest number of recordings that has been used to evaluate makam recognition task, so far. As explained by the authors, these recordings were selected from the performances of "indisputable masters," and therefore they are musically representative of the covered makams. Nevertheless, the results are not reproducible as the dataset is not public.

The tonic identification methodology proposed in [6] was evaluated using 150 synthesized MIDI files plus 118 solo recordings. Similar to [13] the data is not publicly available. To the best of our knowledge, there exists only two open tonic identification datasets for OTMM, both of which are compiled under the CompMusic project.[9] The first one is provided in [22] and it consists of 257 audio recordings. The second and the bigger test dataset is provided in [2], consisting of 1093 recordings.[10] The recordings in both of the datasets are identified using MusicBrainz MBIDs.[11] The authors use a variant of the predominant melody extraction method proposed in [21], which is optimized for OTMM [3]. Then they filter the predominant melody to get rid of the spurious estimations and correct the octave errors as explained in [6]. To the best of our knowledge this procedure [3] currently gives the most reliable predominant melody estimations for OTMM.[12] Nevertheless, the predominant melodies extracted from the audio recordings are not provided in either dataset.

Considering the lack of open test datasets for makam recognition and the drawbacks of the available tonic identification datasets, we have gathered a test dataset of audio recordings with annotated makam and tonic, called the *Ottoman-Turkish makam recognition dataset*.[13] The dataset covers 20 commonly performed makams[14] and it is composed of 1000 audio recordings. Following our constraint in the problem definition, a single makam is performed in each recording (i.e. there are 50 recordings per makam). To the best of our knowledge, our dataset is the largest and the most comprehensive dataset for the evaluation of automatic makam recognition. Moreover, it is comparable to the aforementioned dataset provided in [3] for the evaluation of tonic identification methodologies.

Similar to [2] and [22], the recordings in the dataset are labeled with MusicBrainz MBIDs. The tonic frequency of each recording is annotated manually using the procedure explained in [22] and the annotations are cross checked by at least two annotators. For the sake of reproducibility, we also provide the predominant melodies extracted from the audio recordings. We use the open implementation of the predominant melody extraction methodology explained above [3].

Similar to [13], the dataset is intended to be musically representative of OTMM. To achieve this, we selected the recordings of acknowledged musicians from the CompMusic makam corpus [27], which is currently the most representative music corpus of OTMM aimed at computational research. The dataset covers contemporary and historical, monophonic and heterophonic recordings, as well as live and studio recordings. Some of the recordings have non-musical sections, such as clapping at the end of live recordings, announcements in radio recordings or scratch and hissing sounds throughout the historical recordings. This diversity gives us the opportunity to test the methods in a much more challenging environment, which hasn't been completely addressed in the previous research [13].

5.2 Experimental Setup

In the experiments we use stratified 10-fold cross validation. Table 1 gives a combination of the parameters used in the experimental setup. We use grid search, to find the optimal parameters for OTMM. **(F1)** is selected as the state of the art in predominant melody extraction for OTMM [3]. The parameter combinations where the bin size β **(P1)** is greater than or equal to 3 times the kernel width σ **(P2)** are omitted. We also conduct experiments using the raw distributions, without smoothing. When the training model consists of a "single" distribution per mode, the number of neighbors, k **(P4)**, is always taken as 1 as each label is represented by a single data point. The minimum peak ratio **(P5)** is only used in tonic identification and its optimal value is found separately as will be explained in Section 5.3.

For mode recognition, we mark the classification as $True$, if the estimated mode and the annotated mode for a recording are the same. The tonic octave in the orchestral performances of OTMM is ambiguous as each instrument plays the melody in their own register. Therefore, we aim to identify the tonic pitch class. We calculate the octave-wrapped cent distance between the estimated and the annotated tonic, i.e. $min\Big(\big(|e^{(r)}| \text{ mod } 1200\big), 1200 - \big(|e^{(r)}| \text{ mod } 1200\big)\Big)$, where mod is the modulo operation. Remember that $e^{(r)}$ is the normalization of the estimated tonic frequency e, with respect to the annotated tonic frequency r (Equation 2). If the cent distance is less than 25, we consider the tonic identification as correct. In the case of joint estimation, we require both tonic and mode estimates to be correct.

For each fold we compute the accuracy, which is the number of correct estimations divided by the total number of testing data. In Section 5.3, we report the highest average accuracies of the folds for each parameter combination. For all results below, the term "significant" refers to statistical significance at the $p = 0.01$ level as determined by a multiple

[8] https://github.com/sertansenturk/makam_recognition_experiments

[9] http://compmusic.upf.edu/

[10] The datasets are hosted in https://github.com/MTG/turkish_makam_tonic_dataset/releases/

[11] https://musicbrainz.org/doc/MusicBrainz_Identifier

[12] The code of the methodology is available at https://github.com/sertansenturk/predominantmelodymakam

[13] https://github.com/MTG/otmm_makam_recognition_dataset/tag/v1.0.0

[14] i.e. Acemaşiran, Acemkürdi, Bestenigar, Beyati, Hicaz, Hicazkar, Hüseyni, Hüzzam, Karcığar, Kürdilihicazkar, Mahur, Muhayyer, Neva, Nihavent, Rast, Saba, Segah, Sultanıyegah, Suzinak and Uşşak

Table 1: The summary of the tasks, features, training models and parameters used in the experiments

	Name	Values	Comment
T	task	mode, tonic, joint	
F1	predominant melody, X	[3]	extraction method specialized for OTMM
F2	distribution, h	PD, PCD	
M	type of the training model, M	single, multi	number of distribution per mode used in [10, 13]
P1	bin size, β	$7.5, 15, 25, 50, 100$ cents	
P2	kernel width, σ	"no smoothing" & $7.5, 15, 25, 50, 100$ cents	Combinations with $\beta \geq 3\sigma$ are omitted.
P3	distance or dissimilarity	L$_1$, L$_2$, L$_3$, Bhattacharyya, 1−intersection, 1−cross_correlation	1−intersection and 1−cross_correlation are dissimilarities computed from the namesake similarity measures
P4	number of nearest neighbors, k	$\{1, 3, 5, 10, 15\}$	for the "single" distribution per mode training model, the value is fixed to 1
P5	minimum peak ratio	$[0, 1]$	with a step of 0.5, only for tonic identification and joint estimation

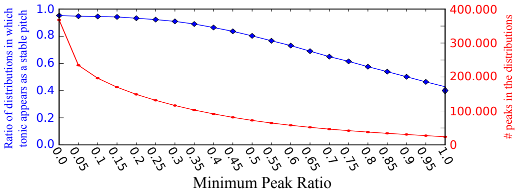

Figure 3: Total number of peaks and the ratio between the number of tonic hits and number of all distributions.

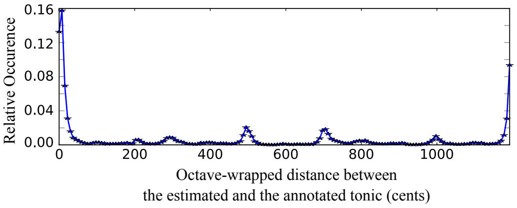

Figure 4: The distribution of octave-wrapped distances between the estimated and annotated tonic for all parameter sets with 7.5 cent bin size.

comparison test using the Tukey-Kramer statistic.

We compare the tonic identification results obtained in the tonic identification and joint estimation tasks with the results obtained from the current state of the art in OTMM [2]. This method is based on detecting the last stable pitch of the recording, which is typically the tonic.[15]

To find an optimal for the minimum peak ratio **(P5)**, we compute numerous distributions of each recording in the dataset using all the combinations of the bin sizes **(P1)** and the kernel widths **(P2)** given in Table 1. Then, we detect the peaks in each pitch distribution using a minimum peak threshold from 0 (no threshold) to 1 (only keeping the highest peak). For each value of the minimum peak ratio, we note the number of distributions which has the annotated tonic among the peaks (”tonic hits”) and the total number of peaks obtained from each distribution.

5.3 Results

By inspecting Figure 3, we observe that the probability of finding the tonic among the peaks is very high for minimum peak thresholds less than 0.4 in the expense of an exponential increase in the tonic candidates (peaks) and hence in the processing time. Since our scenario can tolerate a moderate increase in processing time, we selected the minimum peak threshold as 0.15.

Table 2 shows the best results obtained after grid search. For mode recognition, multi-distribution per mode model yields an accuracy of 71.8% with the best parameter set while highest accuracy using single distribution per mode is 38.7%. For tonic identification multi-distribution per mode performs with accuracy above 95% in 20 parameter sets and above 90% accuracy in 299 parameter sets out of 1440 experiments, where the highest accuracy obtained is 95.8%. Hence, the method is robust to a variety of parameter selections for tonic identification. On the other hand, Single distribution per mode model yields 89.8% accuracy with the best parameter set. For joint estimation the multi-distribution per mode model performs with 63.6% accuracy in the best configuration while single distribution yields 27.6%.

For all three considered tasks, the optimal choices for **P3**, **P5** and **M** turned out to be Bhattacharyya distance, PCD and multi-distribution per mode.

Table 2: Best parameter sets for each task. For all tasks PCDs using Bhattacharyya distance and training multiple distributions per mode gives the best results.

Task	σ	β	**k**	**Accuracy**
Tonic	7.5	15	3	**95.8%**
Mode	25	25	10, 15	**71.8%**
Joint	20	15	5	**63.6%**

The method proposed in [2] is the state of the art for tonic identification in OTMM culture. We evaluated this approach on our dataset and obtained 79.9% accuracy. Multi-distribution per mode method outperforms this method either if the mode is known (95.8% accuracy) or not (91.5% tonic accuracy in joint estimation) with the majority of suboptimal parameter sets. The best tonic identification accuracy using PDs and single-distribution per mode is 49.8. Figure 4 shows the distribution of the octave-wrapped cent distance (Section 5.2) between the estimated and the annotated tonic for each test and for all the parameter sets with 7.5 cent bin size.

These experiments revealed that certain parameter selec-

[15] The open implementation is available at https://github.com/hsercanatli/tonicidentifier_makam

tions significantly improve or diminish the methods' performances. These observations are listed below as a guidance:

- **M:** Multi-distribution training model performs significantly better than single-distribution training model.
- **F2:** PCD significantly outperforms PD.
- **P1:** Smaller bin size yields better results, however there is no significant distinction between 7.5, 15 and 25 cent bin sizes. Note that these bin sizes significantly outperform 50 and 100 cent bin sizes.
- **P2:** The 7.5, 15 and 25 cent kernel widths significantly improves the accuracy of the models compared to 50 and 100 cent kernel widths. No smoothing performs slightly worse than 7.5, 15 and 25 cent kernel widths. However, processing the distribution without smoothing is substantially slower due to the peak detection step.
- **P3:** Using multi-distribution training model and PCDs, Bhattacharyya distance always yields the highest accuracy. It is significant except using $1-$intersection and $L1$ in tonic identification.
- **P4:** Increasing the number of nearest neighbors of k increases the accuracy. Nevertheless, the increase does not make a significant impact except $k = 1$, which performs significantly worse than higher k values.
- **P5:** In the tonic identification task, the true tonic is typically among the detected peaks for minimum peak ratios below 0.4. Values smaller than 0.1 increases the processing time without any meaningful improvement in tonic identification accuracy.

6. EXPERIMENTS ON HINDUSTANI AND CARNATIC MUSIC

Recently, MORTY was used as a benchmark for raga/raag recognition of audio recordings of Hindustani and Carnatic music in comparison with two novel methods [15, 16]. Below we explain the results briefly. Note that there already exists a method for tonic identification for these music traditions [14], which is reported to provide near perfect results. This method is used in both for the automatic tonic identification step. Therefore, the tonic identification and joint estimation using the methods in the toolbox are not applied during these experiments.

For the first of these methods [15], the multi-distribution method was used as the state of the art of the culture. The methods are evaluated for 10 raga and 40 raga setups. The parameters are chosen as $\beta = 10$ cents, $\sigma = 10$ cents, $k = 1$ using Bhattacharyya distance. These experiments were conducted for both entire recordings and the 120 seconds long excerpts. The full recording mode recognition yielded an accuracy of 89.5%, while the windowed excerpts yielded 82.2% in the case of 10 raga scenario and 66.4% and 74.1% in the 40 raga case, respectively.

In the latter work [16], the proposed mode recognition method is applicable to both Hindustani and Carnatic traditions and the multi-distribution approach was again used as the state of the art for comparison. The used Carnatic dataset is composed of 40 ragas and the Hindustani dataset 30 ragas. The results for multi-distribution method was again only reported for the optimal parameter set, which is the same as the aforementioned work. This method performed 91.7% accuracy on Hindustani and 73.1% on Carnatic datasets.

7. DISCUSSION

The drawback of the pitch distribution based methods is that they don't consider the temporal characteristics. When we inspected the results obtained from the experiments in 5, we observed that the confusions are mainly between makams, which either have very similar intervals in their scale or contain similar sets of pitches. Similarly in [16], the proposed method was better in classifying phrase-based ragas, while our method was better at classifying scale based ones. The mode recognition using the feature proposed in [15] is able to capture both of these properties better with a slight increase in computational complexity.

In [2], the authors showed that their method outperforms the tonic identification method in [6] (using PDs with single-model per mode) for OTMM. Our results validate the findings (the best is accuracy is 49.8 as stated in Section 5.3). Nevertheless, we show that using PCDs with multi-model per mode is superior to both methods, even when the makam of the recording is not know and even if the makam is found erroneously in the joint estimation process. While the estimated tonic is typically around the annotation (Figure 4), the main confusion occurs around the fourth, fifth and seventh of the tonic, which typically act as the melodic centers and/or anchor points in the melodic progression [19].

We suggest using multi-distribution models approach with Bhattacharyya distance and PCD. If the estimation accuracy is a top priority, we suggest choosing a small β, σ (7.5 or 15 cents) and minimum peak ratio 0.15 as these parameters yield high accuracies. For use-cases like mobile or real-time applications where computational complexity plays a key role, β, σ (25 cents) and minimum peak ratio (0.4) can be bigger, since reduced feature dimensions substantially decrease the computational complexity. The number of neighbors may be chosen as any value higher than 1.

8. CONCLUSION

We presented MORTY, an open toolbox for mode recognition and tonic identification. The toolbox generalizes the state-of-the-art in pitch distribution based classification for these tasks. It is designed with flexibility in mind such that it can be easily modified and optimized to analyze large audio corpora. We evaluated the implementation on the largest makam recognition dataset of Ottoman-Turkish makam music. Our generalized method outperformed the state-of-the-art methodologies proposed for mode recognition [13] and tonic identification [6, 3]. The toolbox has also been used to benchmark two novel mode recognition methodologies proposed for Indian art musics.

MORTY is also used as a part of our makam music analysis toolbox.[16] Currently it is used to analyze the tuning and obtain a statistical model for each note. We have analyzed the whole audio collection of the CompMusic makam corpus [27]. The automatic description obtained from the analysis is available via Dunya, CompMusic's prototype web-application for music discovery.[17] In the future, we plan to apply dimension reduction and hashing techniques to summarize the features and speed up the classification for real-time mode and tonic estimation on short audio excerpts. We also hope that MORTY may be useful as a general tool for tonic identification, mode recognition and tuning analysis

[16] https://github.com/sertansenturk/tomato

[17] http://dunya.compmusic.upf.edu/makam/

applied on different modal music traditions.

9. ACKNOWLEDGMENTS

This work is partly supported by the European Research Council under the European Unions Seventh Framework Program, as part of the CompMusic project (ERC grant agreement 267583)

10. REFERENCES

[1] S. Abdoli. Iranian traditional music dastgah classification. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 275–280, Miami (Florida), USA, October 24-28 2011.

[2] H. S. Atlı, B. Bozkurt, and S. Şentürk. A method for tonic frequency identification of Turkish makam music recordings. In *5th International Workshop on Folk Music Analysis*, pages 119–122, Paris, France, 2015.

[3] H. S. Atlı, B. Uyar, S. Şentürk, B. Bozkurt, and X. Serra. Audio feature extraction for exploring turkish makam music. In *3rd International Conference on Audio Technologies for Music and Media*, Ankara, Turkey, 2014. Bilkent University.

[4] E. Benetos and A. Holzapfel. Automatic transcription of Turkish microtonal music. *Journal of the Acoustical Society of America*, 138(4):2118–2130, 2015.

[5] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra. Essentia: An audio analysis library for music information retrieval. In *Proceedings of 14th International Society for Music Information Retrieval Conference*, 2013.

[6] B. Bozkurt. An automatic pitch analysis method for Turkish maqam music. *Journal of New Music Research*, 37(1):1–13, 2008.

[7] B. Bozkurt. A system for tuning instruments using recorded music instead of theory-based frequency presets. *Computer Music Journal*, 36:43–56, 2012.

[8] S.-H. Cha and S. N. Srihari. On measuring the distance between histograms. *Pattern Recognition*, 35(6):1355–1370, 2002.

[9] P. Chordia and A. Rae. Raag recognition using pitch-class and pitch-class dyad distributions. In *Proceedings of the 8th International Conference on Music Information Retrieval*, pages 431–436, Vienna, Austria, September 23-27 2007.

[10] P. Chordia and S. Şentürk. Joint recognition of raag and tonic in north Indian music. *Computer Music Journal*, 37(3):82–98, 2013.

[11] P. Dighe, P. Agrawal, H. Karnick, S. Thota, and B. Raj. Scale independent raga identification using chromagram patterns and swara based features. In *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*, pages 1–4, July 2013.

[12] P. Dighe, H. Karnick, and B. Raj. Swara histogram based structural analysis and identification of indian classical ragas. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, November 4-8 2013.

[13] A. C. Gedik and B. Bozkurt. Pitch-frequency histogram-based music information retrieval for Turkish music. *Signal Processing*, 90(4):1049–1063, 2010.

[14] S. Gulati, A. Bellur, J. Salamon, H. G. Ranjani, V. Ishwar, H. Murthy, and X. Serra. Automatic tonic identification in Indian art music: Approaches and evaluation. *Journal of New Music Research*, 43:53–71, 2014.

[15] S. Gulati, K. Ganguli, J. Serra, S. Şentürk, and X. Serra. Time-delayed melody surfaces for raga recognition. In *Proceedings of 17th International Society for Music Information Retrieval Conference*, page (accepted)), New York, USA, 2016.

[16] S. Gulati, J. Serrà, V. Ishwar, S. Şentürk, and X. Serra. Phrase-based rāga recognition using vector space modeling. In *41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2016)*, pages 66–70. IEEE, 20/3/2016 2016.

[17] G. K. Koduri, S. Gulati, P. Rao, and X. Serra. Rāga recognition based on pitch distribution methods. *Journal of New Music Research*, 41(4):337–350, 2012.

[18] G. K. Koduri, V. Ishwar, J. Serrà, and X. Serra. Intonation analysis of rāgas in Carnatic music. *Journal of New Music Research*, 43(01):72–93, Jan. 2014.

[19] I. H. Özkan. *Türk mûsikısi nazariyatı ve usûlleri: Kudüm velveleleri*. Ötüken Neşriyat (in Turkish), 2006.

[20] H. S. Powers, et al. Mode. Grove Music Online, Oxford Music. Online: http://www.oxfordmusiconline.com/subscriber/article/grove/music/43718pg5S.

[21] J. Salamon and E. Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6):1759–1770, 2012.

[22] S. Şentürk, S. Gulati, and X. Serra. Score informed tonic identification for makam music of Turkey. In *Proceedings of 14th International Society for Music Information Retrieval Conference*, pages 175–180, Curitiba, Brazil, 2013.

[23] S. Şentürk, G. K. Koduri, and X. Serra. A score-informed computational description of svaras using a statistical model. In *13th Sound and Music Computing Conference*, Hamburg, Germany, In Press.

[24] S. Shetty and K. Achary. Raga mining of Indian music by extracting arohana-avarohana pattern. *International Journal of Recent Trends in Engineering*, 1:362–366, 2009.

[25] J. O. Smith III and X. Serra. *PARSHL: an analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation*. CCRMA, Department of Music, Stanford University, 1987.

[26] S. M. Suma and S. G. Koolagudi. *Information Systems Design and Intelligent Applications: Proceedings of Second International Conference INDIA 2015, Volume 1*, chapter Raga Classification for Carnatic Music, pages 865–875. Springer India, New Delhi, 2015.

[27] B. Uyar, H. S. Atlı, S. Şentürk, B. Bozkurt, and X. Serra. A corpus for computational research of Turkish makam music. In *Proceedings of the 1st International Workshop on Digital Libraries for Musicology*, pages 1–7, 2014.