A STUDY ON IDENTIFYING MAKAMS WITH A MODIFIED BOLTZMANN MACHINE

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF INFORMATICS

OF

THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

KEMAL TAŞKIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF COGNITIVE SCIENCES

JULY 2005

Approval of the Graduate School of Informatics

Assoc. Prof. Nazife Baykal
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Deniz Zeyrek
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Semih Bilgen
Co-Supervisor

Prof. Dr. Hasan Gürkan Tekman
Supervisor

Examining Committee Members

Assoc. Prof. Ferda Nur Alpaslan	(METU, ENGINEERING)	________________
Prof. Dr. Hasan Gürkan Tekman	(METU, INFORMATICS)	________________
Prof. Dr. Semih Bilgen	(METU, ENGINEERING)	________________
Asist. Prof. Bilge Say	(METU, INFORMATICS)	________________
Dr. Ayşenur Birtürk	(METU, ENGINEERING)	________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Kemal Taşkın

ABSTRACT

A STUDY ON IDENTIFYING MAKAMS
WITH A MODIFIED BOLTZMANN MACHINE

Taşkın, Kemal

M.Sc., Department of Cognitive Sciences

Supervisor: Prof. Dr. Hasan Gürkan Tekman

Co-Supervisor: Prof. Dr. Semih Bilgen

July 2005, 70 pages

Makams are well-defined modes of classical Turkish music. They can be taken as the Turkish music counterparts of Western music tonal structures at a certain level. Nevertheless, makams have additional features such as the usage of specific notes resulting from their different architecture and the special use of scales (i.e. progression). The main goal of this study is to construct a platform for identifying makams through a computer program by proposing a machine learning mechanism. There are restrictions on the mechanism related to the characteristics of the task. Such a mechanism should represent real-time sequential input with continuous values, should handle possible errors in this input and show immediate learning with limited data. These restrictions are valid and necessary for an analogy with the act of listening to music. A Boltzmann machine, modified for this purpose is designed, implemented and used in this study as this learning mechanism. Two characteristics of this study define its significance. First, this study is on the structural features of makams of classical Turkish music. Second, the identifying mechanism is a Boltzmann machine having a different schema than statistical identification tasks in tonality induction.

Keywords: Boltzmann Machines, Classical Turkish Music, Makam

ÖZ

A STUDY ON IDENTIFYING MAKAMS
WITH A MODIFIED BOLTZMANN MACHINE

Taşkın, Kemal

M.Sc., Department of Cognitive Sciences

Supervisor: Prof. Dr. Hasan Gürkan Tekman

Co-Supervisor: Prof. Dr. Semih Bilgen

July 2005, 70 pages

Makamlar klasik Türk müziğinin iyi tanımlanmış modlarıdır. Belli bir dereceye kadar, batı müziğinde bulunan tonal yapıların Türk müziğindeki karşılıkları olarak görülebilirler. Bununla birlikte makamların, kendi yapıları dolayısıyla belirli notaların ve ıskalaların özel kullanımları (seyir) gibi ek özellikleri bulunur. Bu çalışmanın temel amacı bir makine öğrenmesi düzeneği önererek bir bilgisayar programı aracılığıyla makamları tanıyacak bir düzlem oluşturmaktır. Böylesi bir düzeneğin üzerindeki kısıtlamalar ise gerçek zamanlı ve sıralı, olası hataları içeren, reel sayı kümesinden gelen girdileri ifade etmek ve sınırlı sayıdaki öğrenme ve deneme verisi ile çalışma yeteneğine sahip olabilmektir. Bu kısıtlamalar düzeneğin müzik dinlemeyle bir analoji kurabilmesi için geçerli ve gereklidir. Bu çalışma içinde bu amaçla değiştirilmiş bir Boltzmann makinesi öğrenme düzeneği olarak tasarlandı, kodlandı ve kullanıldı. İki karakteristik özelliği bu çalışmayı özel kılmaktadır. İlk olarak, bu çalışma klasik Türk müziğindeki makamların yapısal özellikleri üzerinedir. İkinci olarak ise, kullanılan tanımlama mekanizması istatistiksel tanımlama mekanizmalarından farklı bir yola sahip olan Boltzmann makineleridir.

Anahtar Kelimeler: Boltzmann Makineleri, Klasik Türk Müziği, Makam

ACKNOWLEDGEMENTS

I express sincere appreciation to Prof. Dr. Hasan Gürkan Tekman for his guidance. I also thank Prof. Dr. Semih Bilgen, Dr. Günay Günaydın, Tahir Aydoğdu and Nart Bedin Atalay and for their suggestions and comments. I thank to my family for their never ending faith in me and above all, I must express that I dedicate this study to my grandfather Gavsi Özcan, whose love to music guided me in each step of my research.

TABLE of CONTENTS

LIST of TABLES

LIST of FIGURES

CHAPTER 1

INTRODUCTION

Musical pieces consist of pitches. Throughout these pieces, pitches are not selected at random. There are well-defined structures that define their usage. Moreover, pitches of musical pieces have hierarchical relations among them. Musical pieces usually tend to move towards certain pitches, which are known as the tonal centers (or the tonics in Western music). The problem of finding such tonal centers from notes of musical pieces is known as tonality induction. In past years, tonality induction has been a popular topic in musicology, computer science, psychology and other related disciplines. Various methods of pattern recognition and neurocomputation have been applied to tonality induction with impressive results. Among these methods, Bayesian classification algorithms, self-organizing maps and adaptive resonance theory based networks can be mentioned (Bharucha, 1987; Griffith, 1994; Temperley, 2004). This study is an addition to these studies with two important differences.

First difference is the target music culture. This study is on identifying the tonal structures of classical Turkish music: *makams*. Specifically, it consists of designing and implementing a computer program, which takes lists of values representing fundamental frequency sequences of pitches in musical pieces and returns their makams. The problems to be faced are similar to tonal induction tasks to a certain degree, such as representing event hierarchies or representing tonal neighborhood. However, in order to design a makam identification model, additional dynamics must be included. The origins of these dynamics are the characteristics of this music culture. These dynamics will be summarized in the second chapter.

Second difference is the computational method for the model. A modified Boltzmann machine is used for makam identification in this study. It is modified in order to accept sequential input and have directional connections to represent the act of listening to music in real-time. Boltzmann machines are chosen for the model for several reasons. In general, the main reason is that Boltzmann machines are a form of associative memory and associative memories, specifically Boltzmann machines, have cognitive and biological plausibility. Background on the issue is covered in the fourth chapter. It is worth mentioning at this point that biological plausibility is limited to denote the strength of the analogy between the model and the human neurological system. Similarly cognitive plausibility stands for the strength of the analogy between the model and the act of listening to music. For instance, similarities on presentation of the pieces and on the sufficient number of training pieces for identification are included in such an analogy.

One of the central aims of this study is to design a dynamic network architecture, which depends on purely local information for the problem of tonal induction and do it on classical Turkish music. To what degree this aim is succeeded, will be discussed before the end of the last chapter.

1.1. Outline

The thesis is divided into six chapters.

Chapter 1 (this chapter) is an introduction to the study and its content.

Chapter 2 is an introduction to the classical Turkish music. It includes the principles of this music and presents the elements specific to it. A short introduction to music theory, describing fundamental phenomena related to the study, precedes this introduction. There is also a discussion of two schools on classical Turkish music in this chapter, in order to stress the ambiguity on some of these elements such as genres and progression.

Chapter 3 is the literature survey for the study. Throughout this chapter, studies on tonality induction are referred and a short history of tonality induction is given. Aim of this chapter is to discuss tonality induction and its differences from and similarities to this study.

Chapter 4 is a background and discussion of biological plausibility on associative memories and Boltzmann machines. This chapter summarizes associative memory by referencing Hopfield networks and Boltzmann machines. It is preceded by a short introduction to artificial neural networks.

Chapter 5 is the description of the design of the model. It includes the overall methodology as well as the specific algorithms and representations used in it. Test results for the model are also presented in this chapter.

Chapter 6 contains a discussion on the model, its strength, weaknesses and its cognitive claims, recommendations for future research and a summary of the study.

CHAPTER 2

INTRODUCTION TO CLASSICAL TURKISH MUSIC

2.1. Background on Music Theory

Main elements of music are musical pitches, which are denoted by musical notes. In music, pitches are not used as continuous series. Rather there are discrete steps from pitch to pitch (Dowling & Harwood, 1986 chap. 4). These steps are called intervals and their sizes vary among cultures. Octave interval however, is quite common. The interval between two pitches is an octave, if their fundamental[1] frequencies are in a ratio of two to one. Definition of the octave interval leads to another definition, namely the pitch classes. Pitch classes are pre-defined values representing the tonal hierarchies for scales. They correspond to sets of pitches that have frequency ratios of powers of two.

[1] Pitches are composed of simultaneous vibrations of several components at different frequencies. These frequencies are approximately integer multiples of a fundamental frequency (Justus & Bharucha, 2002), which dominates the perceived pitch.

2.1.1. Tonal Material

Frequency is a continuous parameter; however number of pitches within an octave is finite. A fundamental division of the octave is mentioned by Dowling and Harwood:

> "..the octave should be divided into a series of minimal intervals, all equal in size, which are added together to construct all intervals used in melodic scales." (Dowling & Harwood, 1986 chap. 4).

This division of the octave varies among the cultures drastically. For instance in Western music octave is divided into twelve intervals (semitones), in Arabic music they are divided into twenty-four intervals (quarter tones) and in classical Turkish music the division is to fifty-three[2] (Özkan, 1984 pp. 54-65). In western music, all semitones in an octave constitute the chromatic scales. These tones constitute the tonal material in music (Dowling & Harwood, 1986 chap. 4).

2.1.2. Tuning Systems and Scales

Not all possible tones from the tonal material are used in musical pieces. Instead of this, subsets of these tones are used in each musical piece. These subsets can be represented as certain interval patterns, called (melodic) scales. Theoretically, a scale can be formed by any interval pattern within an octave. However, this is not the case in actual composition. For instance, in Western music there are two types of scales: major and minor. They have the sequence of intervals T-T-S-T-T-T-S, T-S-T-T-S-T-T

[2] Western musicologists also denote music cultures having smaller intervals than their music as microtonal.

correspondingly, where T denoting a whole tone (a frequency ratio of $2^{1/6}$) and S denoting a semi tone (a frequency ratio of $2^{1/12}$). It is worth mentioning that this is not a restriction. In other music cultures situation may be different. For instance in classical Turkish music whole tones are divided to nine equal intervals that are named commas, instead of semi tones that divides wholes tones to two.

2.1.3. Tonal Hierarchies

It is common for each music culture, that pitches in a scale have different priorities; they are not used equally distributed throughout the musical pieces. For instance, the first pitch of the scale (denoted as *the tonic* in Western music) has more influence on the scale than the second pitch of the scale. Moreover, all other pitches of the scale are in some form of a structural hierarchy (Krumhansl, 1990 chap. 4) that defines tendencies and expectancies to certain pitches in scales. Music cultures having such a hierarchy are said to be tonal. Another form of hierarchical relation in music is the event hierarchy. Event hierarchies define the significant usage of pitches in the context. Following a discussion on the tonal hierarchies of north Indian music Bharucha describes the contrast between these two hierarchies:

> “Event hierarchies describe the encoding of specific pieces of music; tonal hierarchies embody our tacit or implicit knowledge of the abstract musical structure of culture or genre. The tone C may occur many times in a musical piece; each occurrence is a distinct musical event. But all the occurrences are instances of a class of tones (tokens of a type) denoted by “C.” In the context of a given piece of music, an event hierarchy represents the functional significance of each occurrence of a C relative to the other sounded tones, whereas a tonal hierarchy represents the functional significance of the class of all C’s relative to the other pitch classes.” (Bharucha, 1984).

To sum up this music theoretical introduction, music is composed of pitches corresponding to certain fundamental frequency values and they constitute the tonal material of music. Well-defined interval patterns defined on these pitches form tuning systems and these tuning systems have certain structures, which are known as scales. Scales are used with hierarchical tonal organizations in music and these organizations vary among music cultures. These hierarchies also have great influence on the perception of music and any music recognition algorithm, independent of its structure, should have a mechanism for them either explicitly or implicitly.

2.2. Classical Turkish Music

Classical Turkish music is thought to have originated from the music of early mid-eastern and Persian cultures; which, in turn, originated from the music of mid-Asian Turks (Yılmaz, 2001 pp. 7-15). Most probably up to 14th & 15th centuries (corresponding to the rise of Ottoman Empire), this music interacted with Byzantine and Arab music and evolved into the classical Turkish music of the present day (Judetz, 1996; Tanrıkorur, 2003a; During & Mirabdolbaghi & Safvat, 1991). These centuries are not the end of the evolution of this music; they are mentioned here since they contain the oldest written musical data having similar characteristics to today's music.

There are a number of characteristic features of this music. First of all, classical Turkish music is not polyphonic. There is a single melody at a single instance of time throughout the piece. The emphasis in this music is the melody, instead of harmony[3]. This is one of the reasons for the fact that classical Turkish music is known as modal (instead of tonal). However, following the notation of Castellano & Bharucha & Krumhansl (1984), it can be argued that this music is still tonal only with a different tonal hierarchy. Counter argument on this claim is that in tonal music, the main problem turns out to be emphasizing the tonal hierarchies by harmony in musical pieces, which contain modulations (changing the tonal center of the piece) among different scales continuously. On the other hand, classical Turkish music is known as modal, because modulation is not the central issue in this music. Second feature to be mentioned in the context is the richness of the usage of time. Usage of time; the usage of rhythmic features is an essential part of this music culture, just as all other music cultures. Throughout the evolution of classical Turkish music, many rhythmic patterns (known as *usûl*) have been introduced. The variety of these patterns constitutes an essential part of classical Turkish music's significance. This part of classical Turkish music is beyond the scope of this study and hence it will not be discussed here further. Detailed information on this issue can be found at Özkan (1984 pp. 557-688).

A third characteristic feature is the usage of pitch, which has certain differences, compared to other music cultures such as Western and Arab music. In its simplest terms, pitch intervals are smaller in terms of consecutive differences of frequencies, providing a basis for the tonal richness of classical Turkish music. There is a huge variety in the usage of tonal structures (Tanrıkorur, 2003b pp. 139-165). Although the basic idea of tonality has many common points with respect to Western music, the usage

[3] Moreover, techniques such as arpeggio and elements such as chords are not used traditionally. The usage of arpeggio in kanun is very lately adopted from Western music and it is still being rejected in many musical forms by the performers.

of scales in classical Turkish music is more flexible in terms of using pitch intervals. It also involves a number of significant sub-structures, which will be mentioned throughout this introduction.

This study is concerned with these tonal structures, leaving usûls and other features untouched; and hence within this perspective tonal dimension of classical Turkish music will be examined here in detail.

2.3. Makams

In classical Turkish music, tonal structures correspond to *makams*. Instead of two different modes (major & minor scales) of Western music, there are more than 30 makams[4] (in use today) in classical Turkish music[5]. Makams define the usage of pitch in this music with their additional structural properties. However, there is not a certain agreement on the ways in which these properties are defined. There are a number of systems, which were followed (and are being followed) and accepted in the history of this music. These systems are Urmevî - Meragî System, Râuf Yektâ Bey School, Arel – Ezgi School and Kantemiroğlu School (Akdoğu, 1996).

---

[4] Makam (*plural makams*) is used in this study. Other commonly used term, mostly in Arab originated texts is maqam *(plural maqâmat)*.

[5] Tanrıkorur (2003b) claims that the actual number is 587, however many of these are today known to be produced for different kinds of social reputation in 18th and 19th centuries, in the late Ottoman era.

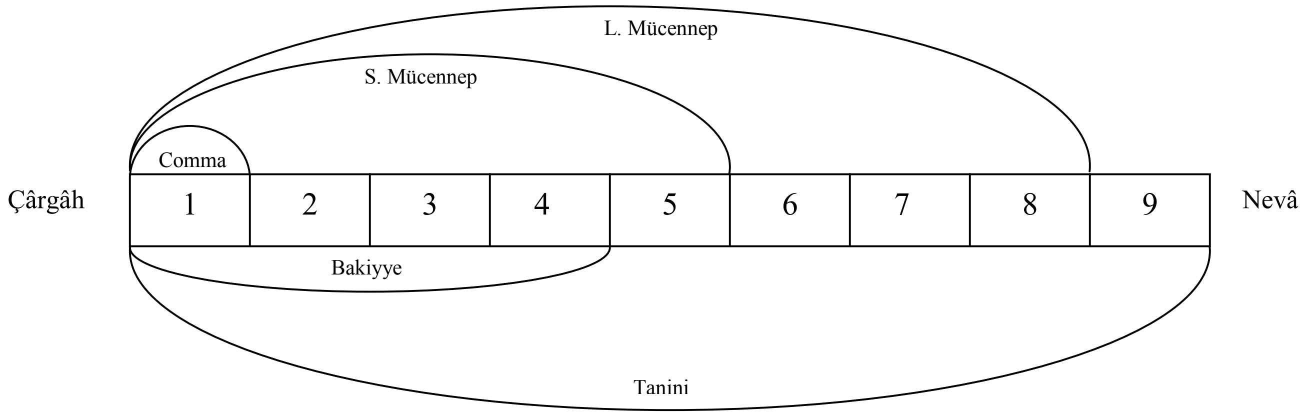
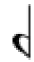
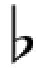
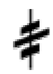
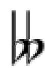


Figure 2.1: Small intervals and their naming in classical Turkish music. In this example, small intervals between Çârgâh and Nevâ are shown. However, there are also 9 small intervals between Nevâ and Acem; Hüseynî and Rast; Rast and Dûgah; Dûgah and Bûselik. There are 4 small intervals between Acem and Hüseynî and Bûselik and Tiz Çârgâh. Complete pitch names can be found at Appendix.

Among all these systems, one can observe that the main differentiation was on determination of intervals and their different nomenclature. However in present day, on determination of (small[6]) intervals and their naming there is a certain agreement (Figure 2.1, Table 2.1). On the other hand, there are different points of view in the definition of makams, especially between Arel – Ezgi School and Kantemiroğlu School.

[6] According to Arel-Ezgi school, there are three types of intervals. A small interval corresponds to 1 comma, a medium interval is either a tetra chord or a penta chord and a large interval corresponds to an octave (Yılmaz, 2001 pp. 37-39).

Table 2.1: Small intervals in classical Turkish music. Another interval is residual duple (A) with a value of 12, 13 or 14 commas. Yet another interval, which is rarely used, is missing bakiyye (E) with a value of 2 or 3.

(Small) Intervals	Symbols	Flat	Sharp	Values
Comma (koma)	F			1 comma
Bakiyye	B			4 commas
Small Mücennep[7]	S			5 commas
Large Mücennep	K			8 commas
Tanini	T			9 commas

2.4. Makam in Arel - Ezgi School

According to this school, there exist medium intervals or genres (a term, which is used for both tetra-chords and penta-chords (Ayari & McAdams, 2003)), which consist of additions of three of four small intervals[8]. These medium intervals constitute the structure of makams: A simple makam is an addition of a tetra-chord to a penta-chord

[7] *Mücennep* (*mücenneb*) is an Arabic word originated from *cenb*, which means to be *side part.* It was introduced to the field by a philosopher and musicologist Farabî.

[8] At this point it is worth mentioning that the term genre will be used for referring to these intervals from this point. A well known meaning attributed to genres is the musical style such as rock or pop, however it is not related to the meaning that is attributed in this study.

(or vice versa) corresponding to a scale having a primary karar[9] note, a secondary karar note (the point of addition), a certain progression and a number of other features (Yılmaz, 2001 pp. 70-79; Özkan, 1984 pp. 93-94). A list of these features and short explanations for them in Arel - Ezgi School is as follows:

Primary karar is the first note of the scale. It is mostly observed at the end of the refrain[10] parts (Figure 2.2).

Secondary karar is the point of addition of the two medium intervals. This is the very important note of the makam and is stressed much throughout the piece. Introduction parts of makams usually end with secondary karar notes.

Other important notes are as follows: *Tiz karar* is the note, which is one octave above the primary karar (tonic). *Asma karar* is specific to a makam. It denotes a note, which is stressed in parallel with the secondary karar. *Yeden* (leading note) is the note one tone (or 5 commas) below the primary karar. It has different effects on makam according to its distance to the primary karar.

Progression is one of the key features of a makam as well in classical Turkish music. There are different makams that have the same scales and same karar notes. However, they are differentiated according to their progressions. For instance Rast and Yegâh both have the same scale (Rast-Dügâh-Segâh-Çârgâh-Nevâ-Hüseynî-Eviç-Gerdâniye), however Rast is an ascending and Yegâh is a descending makam in Arel-Ezgi notation.

[9] Karar is the Turkish word for stability. Hence the primary karar is the primary stability note (tonic), etc.

[10] Turkish word for this is *teslim*, which is an exact translation of *delivery* or *submission*.

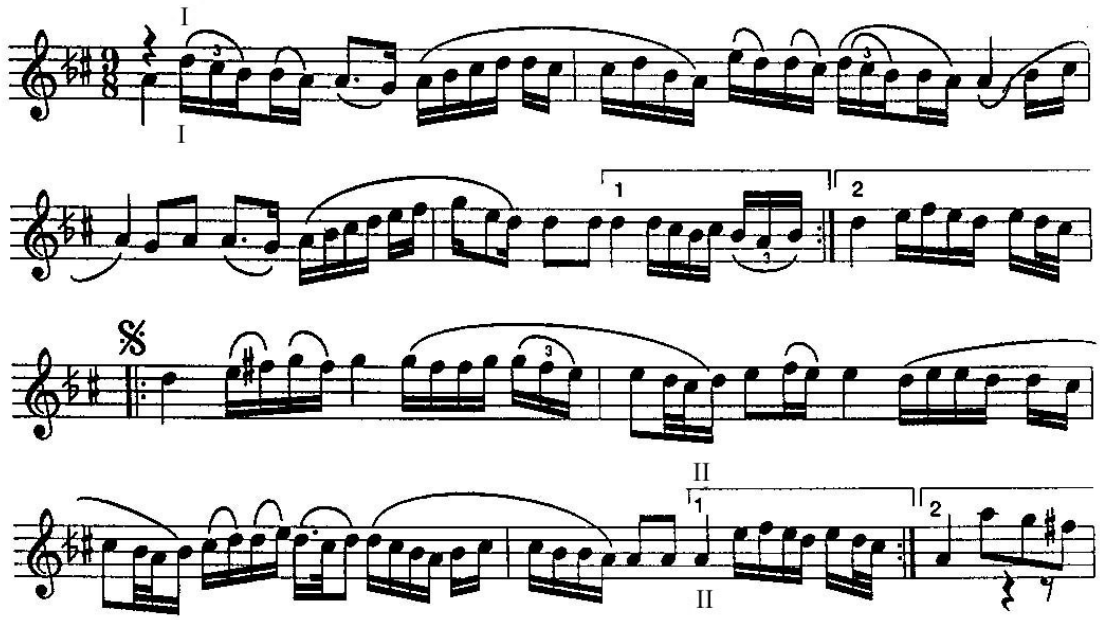


Figure 2.2: Introduction and refraining parts of Hicâz song *Görmedim Ömrümün Âsûde Geçen Demini* by Kadri Şençalar. Hicâz is an ascending-descending makam in terms of progression and hence the piece starts with the secondary karar (denoted by Roman numeral I). The introduction ends with again secondary karar. Refraining part ends with primary karar, independent of the type of the progression (numeral II). Ascending-descending progressions do not have certain beginning phrases; they can also start with the primary karar. Here, the composer denotes primary karar by stressing it as the very first note in the piece. These very first notes take place in classical Turkish music usually to initialize the performance (by a drum).

Arel - Ezgi School classifies progression into three sub-groups: Ascending, descending, ascending-descending. The classification is as follows: In ascending makams the progression starts with the lower tetra-chord or penta-chord, most probably around the primary karar, with a tendency to tiz karar. In descending makams the progression starts around the tiz karar and moves towards the secondary karar. And finally in ascending-descending makams the progression starts around the secondary karar and tends to move towards the primary karar.

There are two more types of makams. Transposed makams are the transpositions of simple makams and compound makams are makams, in which there are compounds of genres in any unsystematic way (with respect to Arel – Ezgi). At this point, it is worth mentioning that this systematization fails to explain many simple makams (according to Urmevî - Meragî System and Kantemiroğlu School) in its conjecture and classifies them as compound makams, just because they are not additions of certain genres.

Table 2.2: Genres in Arel - Ezgi School and their values (ID's are introduced for simplicity in this study).

Tetra-chords	**ID**	**Values**	**Penta-chords**	**ID**	**Values**
çârgâh tetra-chord	I	T-T-B	çârgâh penta-chord	1	T-T-B-T
bûselik tetra-chord	II	T-B-T	bûselik penta-chord	2	T-B-T-T
kürdî tetra-chord	III	B-T-T	kürdî penta-chord	3	B-T-T-T
râst tetra-chord	IV	T-K-S	râst penta-chord	4	T-K-S-T
uşşâk tetra-chord	V	K-S-T	hüseynî penta-chord	5	K-S-T-T
hicâz tetra-chord	VI	S-A-S	hicâz penta-chord	6	S-A-S-T

The genres summarized by this school in the construction of principle makams and these makams are given in the tables[11] 2.2 and 2.3.

Table 2.3: Simple makams, represented as additions of genres.

Makams	**Additions**
Çârgâh	1+I
Bûselik	2+VI
Kürdî	III+2
Râst	4+IV
Uşşâk	V+2
Hüseynî	5+V
Nevâ	V+4
Hicâz	VI+4
Hümâyun	VI+2
Uzzâl	6+V
Zengüle (Zirgüle)	6+VI
Karcığar	V+6
Sûznâk	4+VI

2.5. Makam in Kantemiroğlu School

The important difference between this school and Arel-Ezgi school is mostly related to their different definitions on progression. According to Kantemiroğlu[12], makams are unique structures that can only be explained by their own progressions. All

[11] The notation of Ayari & McAdams, 2003 will be followed throughout the study: Genres begin with lowercase; makams begin with uppercase.

[12] Kantemiroğlu is the name used by Turks for Prince Dimitrie Cantemir, a governor of Boğdan province and a musicologist (Judetz, 1996 pp. 7-11).

other properties of them (primary and secondary karar notes, yeden, etc.) are derived from these progressions.

One of the main differences between Kantemiroğlu and Arel – Ezgi schools is demonstrated in this example: Kantemiroğlu describes makams without referencing to genres. He defines a makam only by its unique progression. This definition of makam also finds strong support in the present day (Karadeniz, 1983). The attributed meanings to progression by these two schools are also different. In Arel – Ezgi School, progression is a feature, which is independent of the context; whereas in Kantemiroğlu School, it is the makam itself. Kantemiroğlu's example to Muhayyer is as follows:

> **(Eng.)** "From its own note (muhayyer) it moves up to tiz-hüseynî and down to aşîrân and returns to dügâh; ending up with a karar. The makam is even a beloved and noble one. Because even if a progression begins from dügâh and an arrival on nevâ takes place and a karar occurs on aşîrân from there, it would only correspond to Nevâ. However, Muhayyer has two special chapters; one consisting of the progression mentioned, which started from hüseynî and a second one in which it continues the progression through all the high-pitched notes coming to hüseynî, stating it exactly and most usually from nevâ it moves down to çârgâh suddenly and fondling sabâ a little it moves to karar note dügâh and finishes the progression affirming itself."
>
> **(Ottoman Tur.) "**Kendi perdesinden tiz-hüseynîye dek çıkar ve aşîrâna değin iner ve gene avdet edüp dügâhda karar-ı istirahat kılar. Makam-ı merkum gerçi ulu ve aziz makamdır. Zira, dügâh perdesinden hareket-i agâze şürû olunsa bile bu nevâya gelinse ve andan avdet olunup aşîrân perdesinde karar kılınsa, safi Nevâ makamı icra olunur. Ve lakin Muhayyer makamının iki fasl-ı mahsusu vardır, biri budur ki, hareket-i agâzesini daima hüseynî perdesinde şüru eyler. İkincisi şudur ki, tiz perdeleri ile tamam-ı hareket eyledikten sonra hüseynî perdesine gelüp ve tamam hüseynî perdesini gösterdikten sonra nevâ perdesini ekseri uçup, birdenbire çârgâh perdesine

> düşer ve biraz sabâ perdesini okşayup dügâh kararına gider, ve anda bittamam kenduyi icra-i beyan eder." (Akdoğu, 1996)

All features defining the makam; primary karar, secondary karar, yeden, asma karar and tiz karar are included in a single description; which Kantemiroğlu calls *the progression*.

Progression phenomenon is a rather fuzzy concept, which will be one of the subtopics to be discussed in later chapters. As a last remark on it, it must be stated that, in the present day, a different and a simpler definition of progression also exists in the literature. A number of musicologists and performers (for instance T. Aydoğdu; a kanun player in Turkey Institution of Radio and Television) simply define progression only with the very beginning and ending phrases of musical pieces. This simpler version of Arel – Ezgi School's progression claims that a makam is descending only if it begins with the tiz karar and tends to move towards the primary karar. Similarly, it is ascending only if it begins around the primary karar and tends to move towards the primary karar again. A similar rule applies to ascending-descending makams, with a difference that they begin around the secondary karar. This point of view again has problems in it. Specifically, the issue seems clear when the makam is descending: available data (musical pieces in literature) are in parallel with it. Pieces begin with (or rarely, around) the tiz karar when their makam is said to be descending. However, the difference between ascending and ascending-descending makams is not so clear. Even observing such pieces will point out this ambiguity, it can be tracked from the Turkish music theory books and writings. In the case of Kürdî for instance, references are in a sort of debate on whether this makam is ascending or ascending-descending. Uşşak and

Hicaz makams are again fuzzy in terms of their progressions (Karadeniz, 1983; Yılmaz, 2001).

As a conclusion to this discussion on makams, it can be suggested that there are two main points of views in the field: Arel – Ezgi and Kantemiroğlu schools. Arel – Ezgi School can be taken as a systematization attempt to classical Turkish music, in the light of the raw explanations corresponding to makams. It is accepted and preferred today mostly for educational purposes. However, as mentioned before, it has still problems such as the existence of genres. Today, a number of musicologists claim that genres do not exist in definitions of makams and they only exist for classification purposes. Kantemiroğlu School, on the other hand, makes a classification only on the primary karar notes and the progression.

Although one can find other issues differentiating these schools, these two points of view are in agreement on many aspects. For instance, according to both schools, a makam is a structure with a well-defined progression of a number of sequential notes having primary and secondary karar notes even though Kantemiroğlu School does not have an explicit remark on secondary karar notes. To sum up, makam has a number of definitions in different schools, having similar as well as different ideas on it. Hence, a task of understanding makams should investigate these similarities, in order to track for the differences.

CHAPTER 3

RELATED WORK

Since there is no significant study on the structural properties or features of classical Turkish music[13], this chapter concentrates on the studies related to this study on two different aspects. One is on makam identification (single study of Ayari & McAdams, 2003) and the other is on tonality induction.

3.1. Perception of Modal Structures in Arab Music

A closely related article to this study is the article of Ayari & McAdams (2003). In this article, the identification and segmentation performances of two groups of listeners (all musicians or musicologists) on a single taksim (in Râst) were compared. Taksim is a well-structured form in classical Turkish music, Persian and Arab music. The steps of a traditional taksim performance are well defined in the context. It is most similar to improvisation of Western music families at a certain level. Nevertheless it is argued that it has some structural characteristics independent of context, most

[13] A study on pitch clustering for musical pieces in classical Turkish music must be counted at this point as an introduction attempt to makam identification by Akkoç (2002).

importantly its relatedness to makams (modes of classical Turkish music): Each taksim is governed by a certain makam (or makams with transitions). One of these groups in Ayari & McAdams' study consisted of Arab listeners and the other group consisted of European listeners. The task was to make segmentations according to the makam being played while listening to the performance.

Results were as expected. Although there were no criteria given for segmentation, it was observed that both groups tended to construct certain strategies. European listeners based their segmentations on the primary and secondary karar notes and one octave higher notes of primary karars. Segmentations according to the development of the phases and the modal changes were also observed. The strategies of Arab listeners on the other hand, were more complex. They tended to segment the piece into modal cells, most of the time corresponding to presentations of certain genres. And from this point they tried to see the whole picture by combining these cells in a hierarchical way. On the other hand, European listeners had difficulty in detecting modal changes. Only some of them detected changes in modal structures; yet they could not identify them.

3.2. Tonality Induction

Makam identification task consists of determining the structural features of the makam of a musical piece from its sequence of notes. Tonality induction, a task that is targeted to harmonic Western music, is very similar to makam identification in this sense. The central aim of tonality induction is to identify the tonal structure (in its simplest terms, tonic) of pitch sequences. Tonality induction algorithms include transitions (modulations) in these sequences except the very early algorithms in the field. Differences between makam identification and tonality induction do exist, mostly related to the differences between two music cultures (namely, classical Turkish music and Western music). Some of these differences are the absence of harmony in the former one and the absence of progression as a distinctive feature in the latter one. However, their purposes are the same: finding the tonal centers of the musical pieces. Hence, it can be suggested that tonality induction studies serve as a background for the task of makam identification.

To begin with, there is a certain psychological background of perception of tonality. Pitches and chords can be perceived as appropriate or inappropriate when followed by a musical segment according to whether they are within the same key with that segment or not (Krumhansl & Shepard, 1979; Krumhansl, 1990 chap. 4). Moreover, melodic sequences that follow tonal patterns are remembered better (Dowling, 1978; Cuddy & Cohen & Mewhort, 1981; Deutsch, 1980). When describing the algorithms in the field, one can see that this psychological background is put into them externally. In other words, a considerable number of algorithms work on the pitch sequences having the information on what a key is and most usually what the characteristics of keys are. For instance they use experimental data in order to calculate prior probabilities or musicological knowledge in order to construct their network structures.

One of the earliest tonality induction algorithm is the general-to-specific concept-learning algorithm of Longuet-Higgins and Steedman (1971). This algorithm scans a musical piece left to right and for each pitch it eliminates all keys whose scales do not contain that pitch. Surely pitches outside the scale cause the algorithm to fail.

However this is not the actual problem of the algorithm since it is possible to improve this algorithm so that it decides according to the best probable key instead of the 100% correct key by applying a method such as version spaces (Mitchell, 1997 pp. 29-36). A number of solutions to this problem are also offered by the authors themselves. For instance if the algorithm fails at the first scan, then it looks at the first pitch of the sequence and decides according to this pitch. The main problem about this algorithm on the other hand is that it returns a single answer for a single piece and can not account for modulations. Vos and Van Geenen (1996) developed this idea further to handle this problem. Their algorithm runs on a sliding window of forty consecutive pitches in a single piece and for each pitch it increases the probability of the keys that contain it in their scales. It can be considered as a naive Bayesian classifier with equal prior probabilities for each key (Mitchell, 1997 pp. 154-199). A more complete tonality induction algorithm based on Bayesian classification method is the algorithm of Temperley (2004). This algorithm finds the most probable tonal centers of segments (of the piece) by finding the most probable structure from the surface of the piece (pitches of the piece):

$$\mathrm{p}(\text{structure} \mid \text{surface}) = \frac{\mathrm{p}(\text{surface} \mid \text{structure})\,\mathrm{p}(\text{structure})}{\mathrm{p}(\text{surface})} \qquad \text{(Eq. 1)}$$

In order to find the most probable structure, one must know the probability of the surface given the structure for every possible structure and prior probabilities for each structure. However, once these probabilities are known, this method finds the best fitting structures to surfaces.

For each instance in a musical piece the structure corresponds to a key, defining the tonal center. For a pitch sequence, Temperley defines prior structure probabilities as the modulations in these keys such that the most probable situation is staying at the same key. For the next step he determines the surface probabilities given the structures. For this purpose he refers to key profiles. Key profiles are likelihood values of twelve pitch classes derived from the chromatic scale of Western music for each twenty-four keys (twelve major and twelve minor).

Even though this Bayesian method gives most successful results in terms of correctness, there is an issue, which is not given an explanation on. Temperley admits that the algorithm works with exponential complexity and it is not feasible in its current form. He also asserts that this problem can be solved by using some kind of dynamic programming techniques, leaving the discussion at this point.

Krumhansl-Schmuckler key finding algorithm (Krumhansl, 1990 chap. 4) is yet another approach to tonality induction. The algorithm correlates twelve-valued key profile vectors to the twelve-valued input vector, which is linearly dependent to the total durations of pitch classes. Finally, the algorithm decides on the key, which has the highest correlation value. The correlation value is defined by the following formula:

$$r = \frac{\sum (x-\bar{x})(y-\bar{y})}{(\sum (x-\bar{x})^2 \sum (y-\bar{y})^2)^{1/2}} \qquad \text{(Eq. 2)}$$

where r is the correlation value, x is the input vector, $\bar{x}$ is the average of the input vector, y is the key profile and $\bar{y}$ is the average key profile. In contrast to the approach of Longuet-Higgins and Steedman this algorithm returns a key at each point in the musical piece by defining x incrementally. İzmirli & Bilgen (1996) is another algorithm, which is similar to Krumhansl-Schmuckler key finding algorithm in this sense. In this algorithm authors propose a method to evaluate the tonal context at any point in the piece. This algorithm also works on the key-pitch relations. In other words, the effect of each pitch on the tonal context is well known.

An enhancement to Krumhansl-Schmuckler key finding algorithm is the algorithm of Toiviainen and Krumhansl (2003). This algorithm proposes a recursive algorithm (on time), passing the pitches sequentially as arguments in a piece. Forgetting and updating mechanisms, which are defined on a pitch memory vector, are defined internally. This memory vector is updated by each presentation of a pitch. This idea that is proposed in this algorithm is similar to the one that will be presented in this study in the sense that both include an implicit memory on pitch sequences.

Another key finding algorithm is the connectionist framework named MUSACT of Bharucha (1987). Bharucha proposes a three-layer neural network, in which the first layer represents pitches, second layer represents chords and the third layer represents keys. These layers have connections in between activating each other so that a pitch is only connected to the chords it is a member of. Algorithm scans the pitch sequence left to right and for each pitch activate the associated node. This node activates the nodes of the second layer and those nodes activate the third layer. With this spreading activation, the algorithm becomes capable of deciding on a key for each instance of this scan. Learning in this algorithm is a competitive one. In other words, connection weights between the layers are adjusted so that activation of one node causes a deactivation of other nodes in a layer.

The last algorithm to be mentioned in this section is the ART (adaptive resonance theory) model of Griffith (1994). ART has a number of applications in different levels of music perception (Griffith & Todd, 1994). In his article, Griffith proposes an ART2 (Carpenter & Grossberg, 1987) network for the purpose of inducing the tonal centers of musical pieces[14].

All approaches, except the ART2 approach of Griffith, implement the sense of key explicitly. In other words, the system knows what a key is by having the information on key profiles or prior probabilities of keys. They decide on a key probabilistically. Because of this reason, they can be classified as statistical approaches. This study is similar to Griffith's approach in this sense. It proposes no additional assumptions such as the key profiles or information about the chords. Such information are stored (implicitly) in the connections of the networks and developed during training. Another similarity between this study and the ART2 approach of Griffith is that both algorithms propose a way of learning to predict the tonality (in case of this study, makams). It is worth mentioning that MUSACT is a connectionist model, however its connections (not the connection weights) and hence the activation schema is pre-defined. Later this approach is preceded by four different unsupervised SOM mechanisms (Tillmann et al., 2000). These mechanisms were hierarchical and similar to the original MUSACT network structure in this sense. They are presented to 12 dimensional input stimuli corresponding to representations of pitches that are present in different chords and expected to classify these different input stimuli. However, this cannot be counted as a feature of MUSACT, rather it is a tautology in the sense that the

[14] Yet another similar ART2 based network is SONNET-1 by Page (1994). However this particular study will not be mentioned here because of its similarity to Griffith (1994).

authors try to verify MUSACT's musicological findings by an externally designed SOM.

There is a similarity between the studies that are presented in this section and the algorithm that will be presented in this study. This similarity is the input representation of the algorithms. All algorithms in this section take the pitches of the musical pieces as input and build their evaluation criteria on this input. The situation is the same in this study, as will be described in the fifth chapter. Another approach is to take the consecutive pitch intervals as input and try to identify interval patterns instead of pitch or pitch-class patterns.

To summarize, these studies do not constitute a complete list of studies in tonality induction (*see* Temperley, 2001 chap. 7; Krumhansl, 2004), however provide insights of considerable importance for this study. Some of these are MUSACT's sequential presentation of pitches onto a finite set of nodes and Toiviainen and Krumhansl's memory for auditory continuity. The actual implementations of these insights are described in the fifth chapter, where the model is presented in detail.

CHAPTER 4

BACKGROUND ON BOLTZMANN MACHINES

4.1. On Artificial Neural Networks

An artificial neural network (ANN) is a set of artificial neurons or nodes connected to each other with artificial connections. These nodes have computational information processing abilities and transfer information through their connections within the network. Information processing mechanisms, network structures, and form and number of connections are the most important characteristics of different network designs.

Artificial neural networks are inspired by the nervous system, more specifically by the neurons and the synapses between these neurons. However, this similarity is not a complete one. There are a number of important differences. First of all, there are a huge number of neurons in the human brain, whereas artificial neural networks are most of the time problem dependent and their sizes are small. Since ANN's are problem dependent (they are designed in order to solve specific problem or for specific problem domains) architectures of these network's nodes are also problem dependent. On the other hand, knowledge on the information processing mechanisms in the human brain is incomplete. Rather than the nervous system as a whole, models are developed for specific portions of this system such as various mechanisms in memory (Amari, 1977; Norman & O'Reilly, 2003), visual object recognition (see Carpenter & Grossberg (1992) for a list of studies) or auditory perception (Grossberg & Govindarajan & Wyse & Cohen, 2004; Hendrik & Blankertz & Obermayer, 2000).

Most common ANN's in the literature are multi-layer neural networks, self-organizing maps, hidden Markov models (HMM), numerous types of associative networks and adaptive resonance theory (ART).

4.2. Associative Memory

Associative memory is a representation, which stands for computational structures that have the capability of storing a finite number of patterns, in a binary form (Gorodnichy, 2001; Kohonen, 1978 chap. 1; Austin, 1996 chap. F1.4). The representation consists of two matrices. The first matrix is one-dimensional and stores class vectors (to be trained and retrieved). The second matrix is two-dimensional and stores values that represent relations among the elements of the first matrix. Storing (the word training is also used) phase of the patterns vary among different structures of the associative memory and specific algorithms used. For instance, these training patterns can be stored directly with a mathematical formula or with a stochastic algorithm. Retrieval (recalling[15], testing) phase, on the other hand, consists of initializing the memory with the input pattern and starting the categorization task. Memory is

[15] In (associative memory) literature the word *recall* is commonly used for this phase. However, *retrieval* will be used in this study for this phase in order to avoid ambiguity.

initialized by setting the units of the memory to values that represent input data. Categorization task involves an algorithm that updates the memory content (the values of nodes) ending in the pattern that resembles to the input most closely. Each memory configuration (list of node values) corresponds to a state in the system. Each state has an energy value defined on nodes and connections. Central aim of the training phase in associative memories is to construct stable states for stored patterns, such that no single change in a node value should be able to decrease the system energy. These stable states are called *attractors*. However, because of the state representations of associative memories are implicit (depend on the values of nodes and connections) false attractors occur in the system, along with the stored patterns. In order to design a computationally robust associative memory, two precautions are necessary in this sense: Designing the training phase in a way that eliminates such false attractors and avoiding getting stuck in them in retrieval phases. One such method for this purpose is simulated annealing, which will be described in later sections.

There are various algorithms designed on the principles of associative memory. Two of them will be described in this chapter because of their relatedness to the study. These are Hopfield networks, and Boltzmann machines[16].

4.3. Hopfield Networks

Hopfield networks are known to be the most commonly used auto associative memory systems. Auto association term implies that input patterns and output patterns are same. Furthermore in Hopfield networks, nodes of the network correspond to both

[16] The focus in this study is on the neural networks; however associative memory phenomenon is not limited with neural network implementations (Gorodnichy, 2001; Neto & Fontanari, 1998).

input and output patterns. In Hopfield network case, nodes of the network correspond to the elements of the binary strings of the patterns.

The standard Hopfield network (Hopfield, 1982) is a fully connected network with McCulloch-Pitts neurons having two states:

$$\xi_i = \mathrm{sgn}(\sum_j w_{ij}\xi_j - \theta_i) \qquad \text{(Eq. 3) (assuming } \theta_i = 0 \text{ for simplicity).}$$

where ξ_i is the state of the i[th] neuron with values {-1,1}, w_{ij} is the connection weight between the i[th] and the j[th] neurons and θ is the loop-back connection (Figure 4.1).

Connection weights are symmetric:

$$w_{ij} = w_{ji} \qquad \text{(Eq. 4)}$$

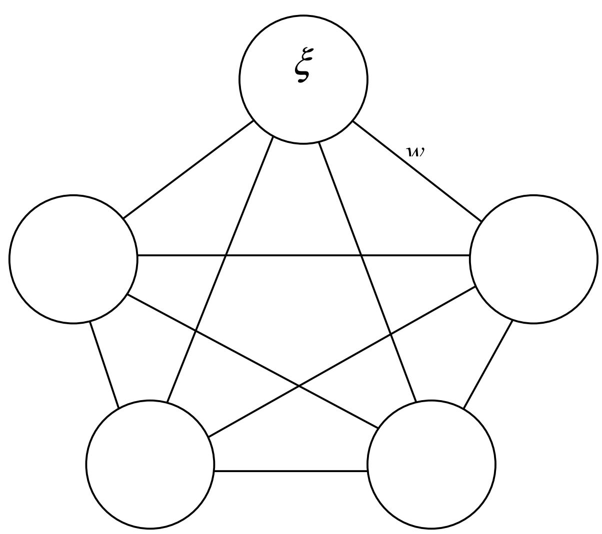


Figure 4.1: Typical Hopfield network. Hopfield networks are fully connected. In this particular figure, the network has five nodes, corresponding to 5 dimensional input and again 5 dimensional output. Since it is fully connected, there are 5*(5-4)/2 = 10 number of connections.

The pattern retrieval algorithm is as follows:

Initialize the network nodes to the input pattern
Until the network reaches to a stable state
    Apply (Eq. 3)
Return the state of the network ξ

Network updating cycle can be designed in a number of ways. In synchronous update all nodes are updated in a single network state in parallel. In asynchronous update, on the other hand, either nodes are selected at random to be updated or they are updated in parallel with some probability.

In a standard Hopfield network, storing patterns is straightforward. It consists of determining the weight matrix of the network with respect to the set of class patterns and a learning rule. This learning rule in Hopfield networks is a form of Hebbian learning:

$$w_{ij} = \frac{1}{N}\sum_{\mu} \xi_i^{\mu} \xi_j^{\mu} \qquad \text{(Eq. 5)}$$

where μ corresponds to each pattern and N is the learning rate, which is problem specific.

A last word on Hopfield networks should be on the convergence of retrieval task and hence the energy function. Hopfield defines an energy function on the network as follows:

$$E = -\frac{1}{2}\sum_i \sum_j w_{ij}\xi_i\xi_j \qquad \text{(Eq. 6)}$$

which decreases monotonically in each update operation, which can be verified through the following equations:

$$\Delta H = -\sum_{j \neq i} w_{ij}\xi_i^{'}\xi_j + \sum_{j \neq i} \xi_i\xi_j \qquad \text{(Eq. 7)}$$

$$\Delta H = 2\xi_i \sum_{j \neq i} w_{ij}\xi_j \qquad \text{(Eq. 8)}$$

$$\Delta H = 2\xi_i \sum_j w_{ij}\xi_j - 2w_{ii} \qquad \text{(Eq. 9)}$$

in which the first term in the equation is negative due to (Eq. 3) and second term is again negative (or zero) due to the structure of the network. Since the energy of the network is monotonically decreasing and E is bounded, the network is guaranteed to reach a stable state, which will be one of the stored patterns[17]. w_{ii} values represent the self-connections of nodes. In almost all implementations these values are set to zero for simplicity.

[17] Only limitation on this process is using an asynchronous updating mechanism.

4.4. Boltzmann Machines

A Boltzmann machine can be considered to be a Hopfield network with disjoint input, output and hidden nodes (units) (Hinton & Sejnowski, 1986; Duda & Hart & Stork, 2001 chap. 7; MacKay, 2003 chap. 43). In contrast with Hopfield networks, input patterns and output patterns are two disjoint sets of nodes. Moreover, there is a third set of nodes other than these two sets, which is known to be the hidden layer (Figure 4.2).

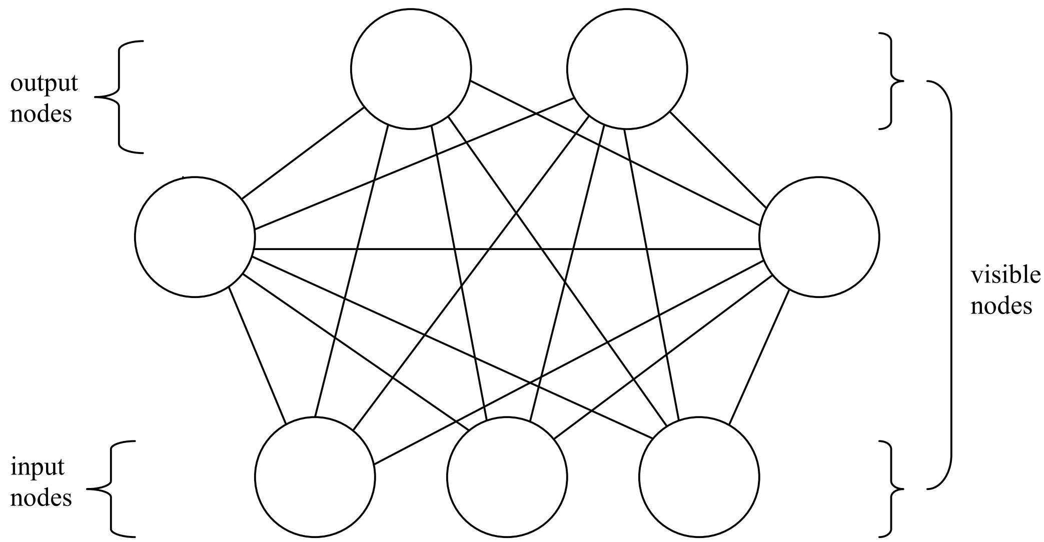


Figure 4.2: A Boltzmann machine with 3 input and 2 output nodes.

Pattern retrieval[18] is similar to the auto-associative memories, however hetero-associative (Davey & Adams, 2002) strategy brings a problem in the Boltzmann machines: Defining the pattern-storing algorithm according to the training set. The problem is that the system cannot be guaranteed to have the stored patterns only. This means that there may appear local minimum points on the energy landscape (that will be

[18] Again it is worth mentioning that another word *test* is used instead of *retrieval* in some related texts.

defined in the following sections), which correspond to false attractors in a Boltzmann machine. The problem is solved with an incremental updating algorithm; however it again leaves local minimum points in the system.

The idea behind Boltzmann machines is to train the network by updating nodes (according to the energy definition of Hopfield networks) with input and output nodes clamped and update the weights according to the differences in the weights with respect to another instance of the machine, which is trained with only input nodes are clamped. Clamping a node, in Boltzmann machines, corresponds to assigning a final value to that node and making it static. After clamping a node the Boltzmann machine is in a situation that it must find the minimum energy configuration with that node having its final value. Network updating on the other hand corresponds to changing the values of the nodes, which will be described with an algorithm at the end of this section.

The Boltzmann machine training algorithm (Deterministic Boltzmann Learning) is as follows (Duda & Hart & Stork, 2001 chap. 7):

Until the network reaches a stable state (a pre-defined convergence criterion met)

> *Randomize nodes*
>
> *Update network with input and output nodes clamped*
>
> *Store node values to* $[s_i s_j]_{\alpha^i \alpha^o \, clamped}$
>
> *Randomize nodes*
>
> *Update network with only input nodes clamped*
>
> *Store node values to* $[s_i s_j]_{\alpha^i \, clamped}$
>
> *Update each weight value*

The weight updating rule is as follows:

$$w_{ij} = w_{ij} + \eta / T([s_i s_j]_{\alpha^i \alpha^o clamped} - [s_i s_j]_{\alpha^i clamped})$$

where η is the learning rate, T is the temperature, s_i is the i^{th} node, $[s_i s_j]$ is $s_i * s_j$ and the notations α^i and α^o correspond to clamped input layer and clamped output layer.

Network updating in this scope is as follows:

Until a pre-defined convergence criterion met

Select a node at random

If energy of the node is greater than zero do

$s_i=(-1)* s_i$

4.4.1. Simulated Annealing

During the network update procedure of the Boltzmann machines, a stochastic mechanism, which is known as simulated annealing, is used (Duda & Hart & Stork, 2001 chap. 7; MacKay, 2003 chap. 43). According to this mechanism, the nodes of the network are updated with a certain probability in order to increase the energy instead of decreasing it. This probability is controlled by a cooling schedule, in which the network update procedure becomes more stable in each step. The metaphor with the temperature phenomenon rises from this point. At a high temperature, system tends to behave in a more randomized manner. There are various cooling schedules in the literature; however their aim is the same: saving the system from getting stuck at local minimum points, which are false attractors (Figure 4.3).

The important point is that the local minima mentioned here are the false attractors among the class patterns. These false attractors exist in the Boltzmann machines and not in the Hopfield networks. The difference from Hopfield networks is that the training phase in Boltzmann machines is stochastic, similar to the retrieval phase (Hinton & Sejnowski, 1986).

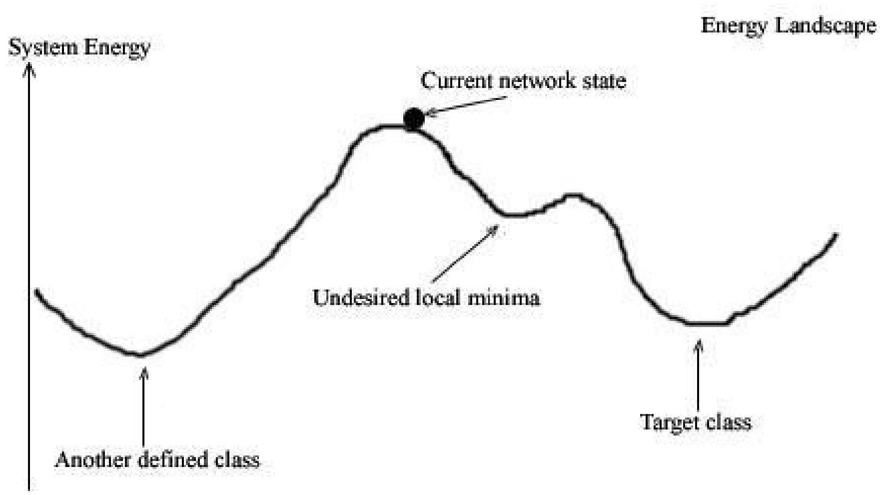


Figure 4.3: Demonstration of the energy space of a Boltzmann machine. As in all associative networks, nodes and connections of the Boltzmann machine represents an energy landscape and each state of the network corresponds to a point on the surface of this landscape.

4.5. Biological Plausibility of Associative Memories

Associative memories have update rules based totally on local information. That is, an update operation of a single node depends solely on its connection weights and the values of the nodes, which are connected to it (the node to be updated) through these connections. It is also thought to be similar in a biological system (Kohonen, 1978 chap.

1). The opposite situation is the existence of a global controller, which will not be able to save the system against dangers such as catastrophic interference[19] (Davey & Adams, 2002).

A second similarity of associative memories with biological systems is on the structure of Boltzmann machines. Boltzmann machines have a very fast learning phase. It stores a pattern in a few number of epochs, whereas the same pattern can be stored in hundreds or thousands of epochs in other supervised learning networks such as multi-layer neural networks.

Boltzmann machines have also an additional biological claim. The claim is the identity of the training and retrieval phases. In both phases, applied algorithm is the same (annealing the system). The only difference is clamping or not clamping the output layer. Identity of the algorithm makes more sense in terms of an analogy between these machines and nervous system, compared to models such as Hopfield networks[20].

There are also studies in neurophysiology related to the topic. Amari (1977) and Little (1974) showed in different studies that biological neurons, which constitute networks, tend to converge to persistent states in the brain.

On the other hand, there are problems considering such biological plausibility. For instance, it must be admitted that Hopfield networks have the problem of being monotonic. For each set of classes there can be generated erroneous input data, with

---

[19] Forgetting old patterns due to the high elasticity of the network.

[20] As a remark, Hopfield networks have similar retrieval schema to Boltzmann machines, however they do not have a training phase at all. Connection values are set directly.

which the network is guaranteed to get stuck at a local minimum[21]. However, as mentioned in this chapter, an associative memory design, namely Boltzmann machines mostly eliminate this problem by applying simulated annealing.

Another question about biological plausibility of associative memories is the spiking behavior of nodes. In the standard designs of all associative memory networks, the nodes can either be set to –1 or 1 and whether this is the situation in the nervous system or not, is of suspect. However, it should be noted that this over-simplification does not imply a limitation on the associative memory design. In this study, a counter example to such an over-simplification will also be presented.

[21] One of the improvements on Hopfield networks is the work of Segura (2001), in which the network behaves stochastic in the learning phase.

CHAPTER 5

A MODIFIED BOLTZMANN MACHINE

A modified Boltzmann machine design for makam identification is described in this chapter. In the first section the cognitive plausibility of the network is discussed. Details of the design and the test results are given in the succeeding sections. Full source code in C# and the data structure are available at http://thesis.kemaltaaskin.net.

5.1. Requirements for an Analogy between a Makam Identification Model and the Act of Listening to Music

Biological (and hence to a certain degree, cognitive) plausibility of associative networks is already discussed in the third chapter. However, an associative network requires a number of additional similarities to the act of listening to music when the problem is classification and cognition of musical pieces. These additional similarities in this particular study are implementing continuous behavior and implementing the sequential presentation of pitches.

5.1.1. Implementing Sequential and Continuous Presentation of Pitches

This property is implemented with the help of a pre-network layer. In this layer, the auditory input-to-fundamental pitch transformation (i.e. pitch extraction) is assumed to be performed and the actual input nodes to the network (corresponding to the sixty-three most common pitches of classical Turkish music) are fired. Sequential presentation is implemented with the continuous values of these nodes and the decay mechanism applied to them. This layer works independent of the core Boltzmann machine. Processes such as the transformation of auditory input are left beyond the scope of this study. However, it is worth mentioning that such improvements and deep analysis of such topics are possible in this model by leaving the other layers and the core mechanism of the model untouched.

Because of this property, this implementation can be seen as a network structure with a dynamic programming technique. The complexity of the algorithm is independent of the piece length and hence the decision procedure can be threaded, i.e. it can run at parallel independent of the presented input data. This means that at any point of the retrieval phase, network has a certain answer to the classification problem. These properties are analogous to the music cognition of human, which makes dynamic programming techniques (and specifically this modified Boltzmann machine) cognitively plausible in this sense, comparing to the relevant statistical methods, which require calculations of probabilities on selected portions of input stimuli (or the entire stimuli).

5.2. Model Design

5.2.1. Overview

Model consists of a core Boltzmann machine and a pre-network input processing system (Figure 5.1). In the pre-network input processing system, the nodes are sequentially set to the fundamental frequency values of the pitches and with certain algorithms provide values to the second layer continuous input (SLCI) and the progression node. Modifying these algorithms is sufficient to improve this simplification of the real life situation (that a listener does not hear pure tones) and leaving the core Boltzmann machine untouched. This is the main reason that this pre-network input processing system is modeled as a separate layer.

Separate from the pre-network system, there are two major modifications on the standard Boltzmann machine. First of these modifications is on the connections. Within the model, there are two unidirectional connections between each pair of nodes (instead of a single omnidirectional node). Hence, the influences of two nodes on each other are not equal. This also provides a basis for asymmetric connections, such that when one of these unidirectional connections is broken, the influence becomes one-way. This is the case in the connections from the progression node to the output layer. The question of mathematical equivalence of such a system to a standard Boltzmann machine is beyond the scope of this study.

Second modification is on the working mechanisms of the nodes in SLCI. In addition to the behavior of standard nodes of Hebbian learning, nodes of this layer have two mechanisms: decaying, self-clamping. Details of these mechanisms will be described in following sections.

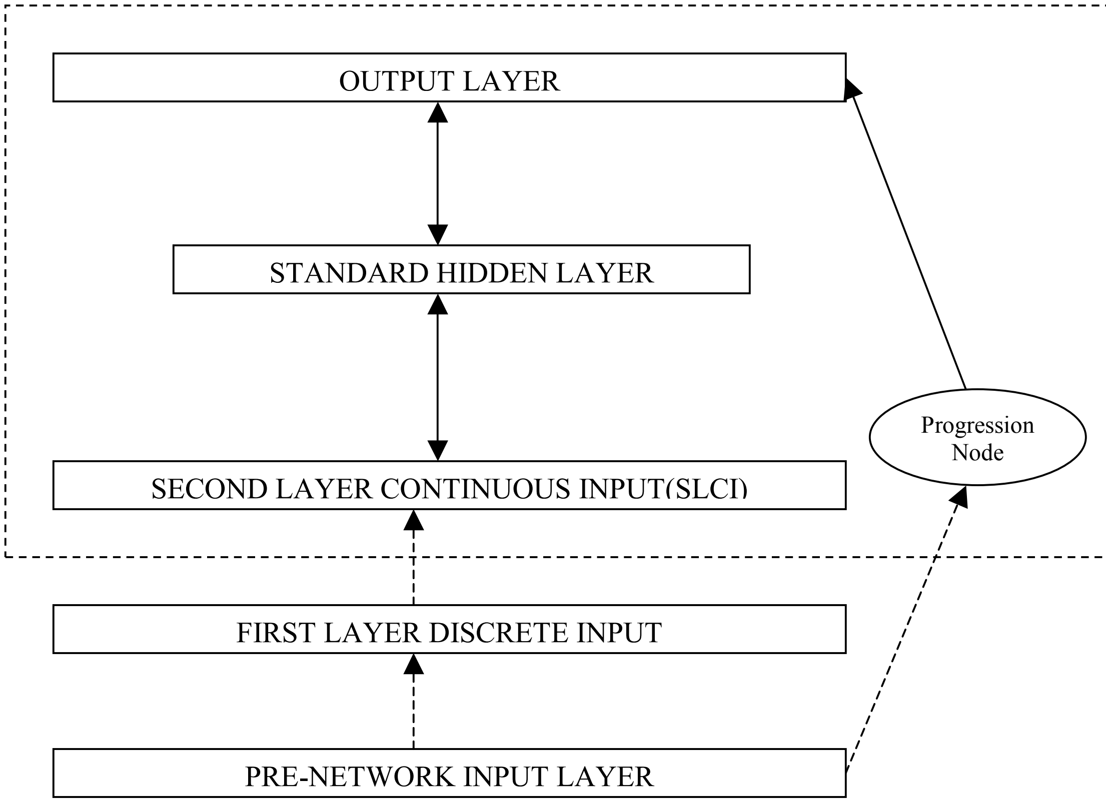


Figure 5.1: Overview of the modified Boltzmann machine. The box with the dashed line is the core Boltzmann machine. Progression node is also included in this box. Directional lines imply one-way connections from each node of the source layer to each node of the destination layer. Bidirectional lines imply similar connections with transitivity (connections in both directions). Dashed lines imply different forms of special connections, which will be described separately.

5.2.2. Pre-Network Input Mechanism

This mechanism consists of pre-network input layer and first layer discrete input. Pre-network input consists of c number of nodes such that all nodes are set to –1 at the beginning of each musical piece. With the presentation of each pitch in the piece, value of the c^{th} node is set to the adjusted value of the pitch presented and values of nodes in the layer are shifted. Value of the i^{th} node is replaced by the value of the $(i+1)^{th}$ one (database design for these representations are also available at http://thesis.kemaltaskin.net).

Adjusting mechanism is a function from frequency values of pitches to [0,1]. The function used in this study is simple as shown in (Eq. 10). Since the model involves a simple transformation from fundamental frequency values to pitches in this layer, this function shows no deficiency. It has no cognitive claims for this moment; however a detailed analysis and improvement on this layer should require a more realistic implementation for this function also.

$$value_{adjusted} = {}^{value_{frequency}}\!/_{2000} \qquad \text{(Eq. 10)}$$

According to (Eq. 10), node values will be in [0,1] since the pitch with the maximum frequency in the pitch database is Tiz Hüseynî having a value of 1319,95.

First layer discrete input has sixty-three nodes, corresponding to the most commonly used pitches in classical Turkish music (Table 5.1). In parallel to the shifting mechanism of pre-network input layer the nodes of the first layer discrete input are updated as follows: The k^{th} node of this layer, which corresponds to the pitch currently being played, is set to 1 and all other nodes are set to –1.

Table 5.1: Complete span of pitches that are used in the simulations. 63 pitches are used in the study beginning from Kaba Yegâh ending in Tiz Hüseynî.

Pitch	Freq.	Pitch	Freq.	Pitch	Freq.
Kaba Yegâh	146,67	Geveşt	371,21	Acem	695,42
Kaba Hûseyni Aşîran	184,99	Dik Geveşt	386,29	Dik Acem	706,06
Kaba Râst	195,57	Râst	391,14	Eviç	732,77
Kaba Zengüle	208,79	Nim Zirgüle	412,04	Mâhur	742,41
Kaba Dügâh	220,00	Zirgüle	417,66	Dik Mâhur	772,58
Kaba Kürdî	231,82	Dik Zirgüle	426,67	Gerdâniye	782,28
Kaba Dik Kürdî	234,87	Dügah	440,00	Nim Şehnâz	824,30
Kaba Segâh	244,26	Kürdî	463,64	Şehnâz	835,15
Kaba Bûselik	247,48	Dik Kürdî	469,87	Dik Şehnâz	868,57
Kaba Çârgâh	260,77	Segâh	488,52	Muhayyer	880,00
Kaba Nim Hicâz	270,05	Bûselik	494,98	Sümbüle	927,28
Kaba Hicâz	274,69	Dik Bûselik	505,23	Dik Sümbüle	939,74
Kaba Dik Hicâz	278,44	Çârgâh	521,54	Tiz Segâh	977,04
Yegâh	293,34	Nim Hicâz	540,10	Tiz Bûselik	989,92
Kaba Nim Hisar	309,03	Hicâz	549,39	Tiz Dik Bûselik	1010,46
Kaba Hisar	313,23	Dik Hicâz	556,88	Tiz Çârgâh	1043,08
Kaba Dik Hisar	320,00	Nevâ	586,69	Tiz Nim Hicâz	1080,20
Hûseyni Aşîran	329,99	Nim Hisar	618,06	Tiz Hicâz	1198,78
Acem Aşîran	347,71	Hisar	626,46	Tiz Dik Hicâz	1113,76
Dik Acem Aşîran	353,03	Dik Hisar	640,00	Tiz Nevâ	1173,38
Irâk	366,38	Hûseynî	659,97	Tiz Hûseynî	1319,95

In the pre-network input mechanism, there is no learning in the connections and the node values are set programmatically. This layer only serves as a feeding mechanism for the SLCI, which can be seen as the actual input to the model. Last function of the pre-network input mechanism is the activation of the progression node, which will be described in the corresponding section.

5.2.3. Second Layer Continuous Input

SLCI has 63 nodes, equal to the first layer discrete input. Each node of this layer is connected to a single node in the first layer discrete input. The relation is one-to-one and onto. In other words, each node of this layer is connected to exactly one node of the first layer discrete input. Whenever a node of this discrete layer is set to 1 the node in SLCI, which is connected to it is also set to 1. Even though the nodes of this layer join the energy function of the core Boltzmann machine, their values are determined from the pre-network input mechanism.

An important property of these nodes is that they can also have continuous values. This is because of the decaying mechanism. In parallel to the node updating algorithm defined within the deterministic Boltzmann machine algorithm, values of these nodes tend to decay. Decaying occurs with a logarithmic function:

$$value(t_{i+1}) = \frac{value(t)-1}{2} \qquad \text{(Eq. 11)}$$

where t represents the instance of time.

The function is logarithmic and this fact causes quick forgetting and decrease in the performance of the network. However, quick forgetting is necessary for robustness. Robustness implies decision procedures, which guarantee producing same result (either correct or wrong) with the same input data. In computational terms, stimuli of the musical pieces are erroneous. Pitches that are out of the current makam are rarely (compared to the pitches of the scale) used in musical pieces. Musical pieces rarely contain pitches only from a single scale. There can be pitches in a musical piece that do not belong to the scale of that piece's makam. Hence, a tonality induction system should have a mechanism to reduce the effect of such pitches in order to be robust in this sense. In other words, a tonality induction system should avoid including these pitches in its evaluation mechanism. This is the reason for using a logarithmic function. However, this brings another problem.

In a standard Boltzmann machine, in both training and retrieval[22] phases input nodes are clamped: they are not subject to updating. However, they are not clamped in this model initially. The reason is to have a point of analogy with the act of listening to music. Before listening to a musical piece, listener is usually not presented to the pitches, their probabilities and sequence. This instead occurs during the listening process. The situation is represented by a mechanism called self-clamping. The idea of self-clamping is as follows: If a pitch is being played too often, corresponding node clamps itself to 1 until the end of the piece. Being too often is represented by the following algorithm:

Clamp the node if corresponding pitch is played more than k times.

where

$$k = \frac{\| notes_{played} \| * (\| notes_{total} \| - 45)}{15 * \| notes_{total} \|} + 3 \qquad \text{(Eq. 12)}$$

[22] Learning (instead of training) and unlearning (instead of testing) words are also commonly used.

and it is a pre-defined algorithm inside the network[23]. Clamping input nodes is a step in training and retrieval phases in a standard Boltzmann machine applied in order to make robust classifications. However in noisy situations such as input vectors having missing values, Boltzmann machine also finds the values of the nodes (nodes that are corresponding to the missing input vectors), since the connections are bi-directional. Starting the phases with all input nodes unclamped and applying self-clamping along with the sequential input is again almost robust because of this behavior of Boltzmann machines. To summarize, logarithmic forgetting and self-clamping gives the model the ability of ignoring out-of-scale pitches without sacrificing the decaying mechanism. However, at this point it must be mentioned it is also possible to increase the performance of self-clamping, a similar mechanism such as *self-unclamping* can be introduced when pitches are no longer played.

5.2.4. Progression Node

Progression layer is designed to be a single node. It is partially out of the core Boltzmann machine. Its value is determined with an externally defined algorithm whereas it joins the energy function of the core Boltzmann machine: there are directional connections from progression node to each node in the output layer.

Value of the progression node is determined by the following algorithm:

[23] Any form of mechanism can be used here. Among them, self-organizing maps and multi-layer networks can be counted. However, the cognitive strength of the overall model would not increase or decrease among any choice for this mechanism. In any case, there would be a need of a pre-defined algorithm. The choice of the current algorithm is arbitrary.

Set direction and coefficient to zero

Starting from the current pitch, trace back to n pitches. Do for each pitch

If the frequency value of the pitch is less than its successor

Decrement direction

Else

Increment direction

If direction is greater than zero

Set coefficient to 1

Else

Set coefficient to –1

Update progression node

where updating rule is

$$value = coefficient * (1 - 0{,}04 * direction) \qquad \text{(Eq. 13)}$$

and value is the value of the progression node. Multiplier coefficient 0.04 is a design choice and depends on the number of total nodes in the core Boltzmann machine. This value is chosen after a number of arbitrary tests. At this point the only remark for these tests is that greater coefficients make this value divergent or reduces its effect to the nodes that it is connected.

Progression node is partially in the core Boltzmann machine, considering this algorithm. The main reason behind this design issue is the fact that progression is defined locally on the pitch sequence. The conclusion is that given a sequence of a number of pitches there exists a decision mechanism whether the sequence is ascending or descending. The phenomenon is known as pitch contour and it is shown that even human infants are sensitive to contour (Trehub & Trainor, 1993). Hence, defining the value of the progression node externally does not decrement the model's plausibility. It should be noted that better representations for progression or better algorithms for determining its value than (Eq. 13) may also be found in this sense. The only criterion for such representations (or algorithms) is to give a tendency to the value of this node to 1, if the pitch sequence has an ascending progression and –1, otherwise. As a corollary, it will be dominated by neither 1 nor –1 and hence the network energy will be independent of it when the pitch sequence has an ascending-descending character.

5.2.5. Hidden Layers and Output Layer

In a standard Boltzmann machine, there is a single hidden layer consisting of nodes connected to each other as well as to the input and output layers. The design of this modified Boltzmann machine is similar to the standard one in this sense.

Output layer consists of nodes, each representing a single makam. In this layer, a form of competition is implemented such that clamping this layer in the training phase corresponds to clamping the node of the makam of the musical piece to 1 and others to –1. In general, the clamped values correspond to the stored classes, defined in the scope of the particular problem. Number of nodes in this layer is implemented to be parametric, however set to 5 in the current study, since the input musical pieces are from 5 different makams: Mâhur, Hicâz, Kürdîli Hicâzkâr, Hüzzâm and Nihâvend.

5.2.6. Overall Training and Retrieval Algorithm

Modified Boltzmann machine algorithm involves a number of modifications on the deterministic Boltzmann machine algorithm, described in the third chapter. However, the characteristics of the algorithm remain unchanged.

Remark on the deterministic Boltzmann machine algorithm:

Until the network reaches a stable state (a pre-defined convergence criterion met)

Randomize nodes

Update network with input and output nodes clamped

Store node values to $[s_i s_j]^{i\ o}_{\alpha\ \alpha\ clamped}$

Randomize nodes

Update network with only input nodes clamped

Store node values to $[s_i s_j]^{i}_{\alpha\ clamped}$

Update each weight value

This algorithm can be modified to have twin networks, in which one of them is used for training, and the other is for retrieval:

Until the network reaches a stable state (a pre-defined convergence criterion met)

Randomize training and retrieval networks

Update training network with input and output nodes clamped

Update retrieval network with only input nodes clamped

Update each weight value according to the differences between nodes of two networks.

The next modification on the algorithm is related to the sequential presentation of pitches in the input layer. Deterministic Boltzmann machine must be modified such that it must be run in another loop, which represents the pitch sequence. Illustration of this modification is as follows:

Randomize training and retrieval networks

Initialize training and retrieval networks

Until the pitch sequence ends, do for each pitch

Feed network with the pitch

Until a number of epochs, which depends on the length of the pitch linearly, do

Update training network with output nodes clamped

Update retrieval network

Apply decaying and self-clamping processes

Update each weight value according to the differences between nodes of two networks.

For a single training stimulus (a sequence of pitches, corresponding to a musical piece with a certain makam) the algorithm starts by randomizing and initializing the networks. Initializing networks corresponds to clamping the output layer[24] for training network. Feeding network with pitches corresponds to update the pre-input network mechanism with pitches and updating the second layer continuous input.

Updating networks by updating the connections between the nodes and applying decaying and self-clamping procedures in the second layer continuous input are the steps performed for each pitch.

After the whole sequence of pitches is presented, weights are updated with a similar rule of the deterministic Boltzmann machine and training is finished for a single piece.

[24] And clamping the genre subset, if being tested.

5.3. Test Results

Testing[25] procedure applied in this study is as follows: The available data are separated into two disjunctive groups. One of these groups is used for training and the other is used for retrieval. Available data involve particular representations of musical pieces from five makams (Table 5.2). Makam selection is arbitrary. However, training and retrieval pieces are selected such that they include the least number of transitions to other makams.

Pieces in the training group are fed to the network for training, each for once (Table 5.3). Following this phase, pieces in the retrieval group are given to the network for testing, each for a single test. Test results are given in tables 5.4 and 5.5.

Table 5.2: Distribution of pieces among makams.

Makam	**Prim. Karar**	**Sec. Karar**	**Yeden**	**Genres**	**Progression**	**# of training pieces**	**# of test pieces**
Mâhur	Râst	Nevâ	Geveşt	1-I	Desc.	2	5
Hicâz	Dügâh	Nevâ	Râst	VI-4	Asc.-Desc.	2	9
Hüzzâm	Segâh	Nevâ	Kürdî	4-VI	Asc.-Desc.	2	7
Kürdîli Hicâzkâr	Râst	Çârgâh	Acem Aşîran	III-2	Desc.	2	5
Nihâvend	Râst	Nevâ	Irâk	2-III	Asc.-Desc.	2	9

[25] Testing in this sentence implies the verification of the model, not the specific testing algorithm applied.

Table 5.3: Training pieces and their makams. Composers of the pieces are given in parenthesis.

Piece Name	Makam
Mâhur Peşrev (Rauf Yektâ Bey)	Mâhur
Sabah Olsun Ben Şu Yerden Gideyim (İ. Ağa)	Mâhur
Hicâz Mandıra (Anonymous)	Hicâz
Aşkı Seninle Tattı Hicranla Yandı Gönül (F. Tokay)	Hicâz
Aşkınla Yanan Kalbimi Sahralara Attın (T. Er)	Hüzzâm
Kadere Bak (A. Şensoy)	Hüzzâm
Girdim Yârin Bahçesine Gül Dibinde Gülizar (O. N. Akın)	Kürdîli Hicâzkâr
Derbeder Bir Aşıkım Yurdum Evim Virânedir (Z. Duygulu)	Kürdîli Hicâzkâr
Nihâvend Yürük Semâi (Tanburi Ali Efendi)	Nihâvend
Nihâvend Peşrev (H. S. Arel)	Nihâvend

Table 5.4: Correct test result counts by makams.

Makam	Correct Test Results
Mâhur	5 out of 5
Hicâz	6 out of 9
Hüzzâm	7 out of 7
Kürdîli Hicâzkâr	4 out of 5
Nihâvend	7 out of 9

Table 5.5: Test results by pieces.

Piece Name	Makam	Result
Mahur Şarkı (N. Kökdeş)	Mâhur	Mâhur
Hicaz Peşrevi (G. G. Han)	Hicâz	Hicâz
Hicaz Saz Semaisi (F. Karamahmudoğlu)	Hicâz	Hicâz
Şarkımı Senin İçin Yazdığımı Bilseydin	Mâhur	Mâhur
Hicaz Saz Semaisi (M. Erev)	Hicâz	Mâhur
Gönül Verdim Bir Civane (Selim, III)	Hüzzâm	Hüzzâm
Manolyam (Z. Müren)	Kürdîli Hicâzkâr	Kürdîli Hicâzkâr
Ayrılık Ne Demek (S. Nevruz)	Hüzzâm	Hüzzâm
Hicaz Şarkı (Şevki Bey)	Hicâz	Hicâz
Kürdîli Hicazkar Peşrevi (T. Cemil Bey)	Kürdîli Hicâzkâr	Kürdîli Hicâzkâr
Adalardan Bir Yar Gelir Bizlere (V. A. Arsoy)	Hicâz	Hicâz
Sazlar Çalınır Çamlıcanın Bahçelerinde (V. A. Ersoy)	Hicâz	Mâhur
Muftunun Oldum (A. Mithat Efendi)	Kürdîli Hicâzkâr	Kürdîli Hicâzkâr
Yüce Dağdan Esen Rüzgar (S. Pınar)	Mâhur	Mâhur
Bağlandı Gönül Zülfüne Bir Yosma Civanın (S. Ahmed Efendi)	Hüzzâm	Hüzzâm
Zülfün Görenlerin Hep Bahtı Siyah Olurmuş	Nihâvend	Nihâvend
Nihavend Şarkı (Rıfat Bey)	Nihâvend	Nihâvend
Nihavend Peşrev (R. Fersan)	Nihâvend	Nihâvend
Çamlarda Şafak Rengi Gibi (V. A. Ersoy)	Nihâvend	Hicâz

Table 5.5: Test results by pieces (continued)

Piece Name	Makam	Result
Güzel Bir Göz Beni Attı Bu Derin Sevdaya (O. N. Akın)	Nihâvend	Nihâvend
Dün Gece Saz Meclisine (M. Sabahattin)	Nihâvend	Mâhur
Nihavend Saz Semaisi (G. G. Han)	Nihâvend	Nihâvend
Hüzzam Peşrevi (T. Osman Bey)	Hüzzâm	Hüzzâm
Sen Bilmiyordun (T. Er)	Nihâvend	Nihâvend
Şu Göğsüm Yırtılıp Baksan (C. Çağla)	Hüzzâm	Hüzzâm
Gökyüzünde Yalnız Gezen Yıldızlar (T. Alpay)	Nihâvend	Nihâvend
Mah Yüzüne Aşıkanım (İsmail Dede Ef.)	Hicâz	Hicâz
Karanfil Türküsü (Anonymous)	Hicâz	Hicâz
Niğde Ninnisi (Anonymous)	Hüzzâm	Hüzzâm
Kuşak Türküsü (Anonymous)	Hüzzâm	Hüzzâm
Bağdadın Hamamları (Anonymous)	Mâhur	Mâhur
Yarim Elimden Gitti (Anonymous)	Hicâz	Mâhur
Koparan Sinemi Ağyar Elidir (B. Şen)	Kürdîli Hicâzkâr	Nihâvend
Bir Kendi Gibi Zalimi Sevmiş Sanıyormuş (L. Atlı)	Kürdîli Hicâzkâr	Kürdîli Hicâzkâr
Ne Doğan Güne Hükmüm Geçer (M. N. Selçuk)	Mâhur	Mâhur

5.4. A Corollary on Makam Neighborhood

Boltzmann machines decide on the principle of system energy, i.e. they reduce their energy to the possible minimum value in the presence of the input stimuli by a certain energy reduction algorithm. That is how they make a classification. In the minimum energy state the output layer represents the correct classification and any change made on the values of this layer will increase the system energy.

As a corollary it can be stated that, with an input stimulus fed to network and the output layer is clamped with the correct classification, the system will reach the minimum energy state after the energy reduction algorithm.

In the current model makams are represented in the output layer with 1 in their corresponding node and -1 in the other nodes. For example, Hicâz corresponds to [-1,1,-1,-1,-1] and Hüzzâm corresponds to [-1,-1,1,-1,-1]. Given such a configuration and the model, a neighborhood among the classes can be established by a set of input stimuli. The idea is to present the input stimuli (musical pieces) with the modified Boltzmann machine algorithm (as described in section 5.2) and then to calculate the system energy for all possible[26] output layer configuration by clamping the output layer accordingly. The complete algorithm for this procedure with a single piece is as follows:

[26] Possible configurations that represent classes.

For a single piece do the following for each makam:

Randomize retrieval network

Initialize retrieval network by clamping the current (neighbor) makam to 1

Until the pitch sequence ends, do for each pitch

Feed network with the pitch

Until a number of epochs, which depends on the length of the pitch linearly, do

Update retrieval network with output nodes clamped

Update retrieval network

Apply decaying and self-clamping processes

Store the energy of the network as the neighborhood value of the neighbor makam

Sort the values and return the list

Then the neighborhood relation will correspond to the average neighborhood of the energy values. However, it is obvious that such a corollary will require a more complete data set, from greater number of makams; which is not the case in this study.

Theoretically, this algorithm sorts the neighbor makams according to the similarities of makams such as the number of common pitches in the scales and the identity of the progression. In practice, one of the most common realizations of makam neighborhood is transition. A transition is a move from one makam to another within a piece. Hence a possible method for verifying the results should be to extract the transitions in the training pieces and compare them with the results drawn from the model. Another idea should be to ask experts which makams are closer to each other (in terms of neighborhood). However, verification of the makam neighborhood claim is beyond the scope of this study and is left as a topic for future studies.

CHAPTER 6

DISCUSSION

This study is on the design and implementation of a Boltzmann machine for the particular task of makam identification. For this purpose the standard Boltzmann machine, which is described in the fourth chapter is modified. The modifications on this model are described in the fifth chapter. Preliminary test results are also given in the same chapter. It is obtained from these results that 29 of 35 test pieces from five different makams are classified correctly; corresponding to a %83 success rate.

Deficiencies of the model on the other hand, are as follows:

The results are far above the chance level (considering %20 for 5 makams), however they can be taken as successful *except* the pieces in Hicâz. If the pieces in Hicâz are removed, the correctness rate equals to %88 representing 23 correct classifications from 26 pieces. The only explanation that will be given here for this (Hicâz) failure will be a number of facts related with this makam. First of all, Hicâz is a makam[27], which has no specific progression characteristic. Pieces in Hicâz can be ascending, descending or ascending-descending. A second fact is that this makam uses Eviç in ascending portions (of pieces) and Acem in descending portions, which increases the system energy by reducing the number of clamped nodes in the second layer continuous input. The last fact to be mentioned is that in Hicâz pieces, Dik Hisar is used almost with the same frequency of Hüseynî, which is the regular pitch of the scale of this makam. This fact also increases the system energy due to the same reason.

These problems can definitely be taken as reasonable facts for such a failure; however it must be noted that in real life such a problem related to Hicâz does not exist. In contrast, Hicâz is known as one of the most popular makams today and it can be suggested that identification of this makam in real life is even easier because of a number of reasons, including the difficulties of the model related to this makam. For instance, frequent usage of Eviç along with Acem is a difficulty for this model; on the other hand it makes identification easier in real life. Another distinctive property of Hicâz is the 13-comma interval between Dik Kürdî and Nim Hicâz. Hence it can be concluded that this modified Boltzmann machine cannot account for this problem and there may be a need of improvements that will unify with these characteristics of Hicâz. For instance, in order to represent an unstable character in the progression layer, this layer can be re-designed to have two nodes or a single node by a pure continuous valued algorithm instead of the previously proposed algorithm (Eq. 13).

Another set of problems arises from the implementation details on the structure of the model. In the modified Boltzmann machine, as well as the standard one there are

[27] Formally Hicâz is not a name of a single makam. It denotes a makam family involving (generic) Hicâz, Hicâz Hûmâyûn, Uzzâl, and Zirgûleli Hicaz. However, in the study generic Hicâz is referred by the word *Hicâz*.

a number of random variables. These variables are the count of nodes in the hidden layer (H) and the connection weight update coefficient (α). There are certain strategies for specifying these variables (Duda & Hart & Stork, 2001 chap. 7). The problem is, the optimum values for these variables change over the number of input and output nodes and the number of connections. Hence, these variables must be adjusted if for example to classify 10 makams instead of 5. A solution to this problem is to design the Boltzmann machine with a greater and fixed number of nodes in its output layer and adjust its parameters accordingly.

6.1. Future Work

One of the key outcomes of this study is that it brings a considerable number of problems besides its success on the test data. It is shown that identifying makams with a modified Boltzmann machine is possible by applying a new mechanism called self-clamping on sequential input data. This sequential input data is a simplified abstraction of actual auditory input, in which the fundamental frequency values of pitches are used. Moreover, because of such a simplification, pre-network input mechanism becomes a straightforward mapping device from the sequential input data to the second layer continuous input. However, it is possible to implement a spectral analysis mechanism in this layer and improve the model in this sense without modifying the core Boltzmann machine.

A second improvement on the model should be on representing the intervals between pitches. Such a representation does not exist in the current model because nodes of the second layer continuous input contain no geometric relation among themselves. In other words, no information about the fundamental frequencies of pitches exists in this layer. This is a serious problem for the model in terms of the strength of its analogy with the act of listening to music. Within this model, which has no representation for pitch intervals, it is not possible to classify transposed makams[28] correctly. The problem is that, the classification is done according to the pitches alone (in addition to the progression node), which are represented topologically. Because of this problem a simulation on the existence of genres cannot be done on this representation. Moreover, a mechanism that can represent pitch intervals may increase the performance of the algorithm on Hicâz because of its characteristic 13-comma interval between Dik Kürdî and Nim Hicâz. A possible solution to this problem within the makam identification context is to design a Boltzmann machine with a different input representation such that the input layer to the core Boltzmann machine contains the information of scale patterns, a limited number of reference pitches and the information of progression. Such a representation will also give better insight on the existence of genres. As mentioned before, it is not possible to have a clear answer to this problem with this modified Boltzmann machine. However, representing interval patterns on the input layer will also represent genres within the same layer and require no additional representations other than the regular hidden layers of the standard Boltzmann machine. For instance, in terms of observing the existence of genres, (nodes of) the hidden layer will become suitable for tracking their activations in order to search for specific activations for specific interval patterns.

[28] In classical Turkish music, makams are usually transposed (pitches of the scale are shifted, leaving the pitch intervals constant) most of the time in order to unify with the singer's vocal range. The ambiguity rises because of the Arel-Ezgi notation. Makams, having the same pitch intervals in their scales have different progressions. In contrast with the Arel-Ezgi school, there is no transposed makams in classical Turkish music. When makams are transposed, they still remain the same makam actually. For instance in Karadeniz (1983) there is no reference to such makams. He refers only to simple and compound makams.

Considering the 12 basic genres of Arel-Ezgi school, further studies can be proposed. A future work on their existence can be to design an experiment on their perception. An experiment, in which these genres are presented to the participants along with some randomly generated interval patterns, should give further insight on the issue. The question of whether these 12 basic genres have perceptual advantages or not, will be revealed with such an experiment.

Another topic for future work is to generalize the model for polyphonic music in order to apply it to Western music and to compare it with the previous tonality induction algorithms. In its current form, the model can take only one input stimulus at a time. However, it is also possible to modify it further for polyphonic music by duplicating the pre-network input mechanism and its connections to second layer continuous input such that each pre-network input mechanism represents one melody. Such a duplication of the pre-network input mechanism will require an assumption that the Boltzmann machine is able to discriminate the melodies into different channels before taking them as input. One related research is on harmonizing musical pieces by Boltzmann machines (Bellgard & Tsang, 1994), which proves that the analogy with the energy function of these machines can work with polyphonic music.

Progression determination algorithm can also be improved as a future work. For this purpose, experiments on perception of contour can be reviewed or new experiments can be designed. The main purpose of such an investigation will be to identify the relation between progression and contour.

Finally, an important topic for future work is to track the system energy in order to identify makams and track modulations throughout the pieces. For this purpose, the output layer of the modified Boltzmann machine can be monitored for each element of the pitch sequence.

6.2. Conclusion

Main purpose of this study was to present a makam identification scheme by proposing Boltzmann machines as the computational model and classical Turkish music as the target music culture. Despite the problems it has, the idea of a modified Boltzmann machine for makam identification is promising. Proposals are given for the deficiencies of the model in the previous section as future work. Yet the idea is promising since it provides a mechanism, which is cognitively plausible to a certain degree. In other words, it proposes an analogy with the act of listening to music. Besides the biological plausibility of associate memories, model provides mechanisms for accepting sequential input, (computational) identity of training and retrieval phases, independence of piece lengths in terms of computational complexity, quick learning (with a plausible and faithful number of pieces compared to networks such as the one in Todd's (1989) approach or multi-layer networks in general) and plasticity and evaluating makam neighborhoods. Analysis on issues such as tracking contour and evaluation of self-clamping are, on the other hand, are left untouched. The algorithms that are provided for these issues are shown to be sufficient for this study in particular; however they are subjects to future work.

This study could also be counted as an application of associative memories on tonality induction, despite the fact that tonality induction has a number of differences from makam identification.

To summarize, this study was an attempt to identify makams from fundamental pitch frequencies and was the first such study with Boltzmann machines and certain perceptual limitations applied. The important outcome is that it is observed that Boltzmann machines can be modified to be suitable for such a task and makam identification can be a well-defined task in machine learning. I believe that further investigations and studies on this particular topic will deepen the knowledge on makams and hence classical Turkish music.

REFERENCES

Akkoç, C. (2002). Non-deterministic scales used in traditional Turkish music. Journal of New Music Research, 31, 285-293.

Amari, S. (1977). Neural theory of association and concept formation. Biological Cybernetics, 26, 175-185.

Akdoğu, O. (1996). Türk müziğinde türler ve biçimler (Types and forms in Turkish music). İzmir: Ege Üniversitesi Yayınları.

Austin, J. (1996). Associative memory. In E. Fiesler & R. Beale (Eds.), Handbook of Neural Computation. New York: Oxford University Press.

Ayari, M. & McAdams, S. (2003). Aural analysis of Arabic improvised instrumental music (taqsĭm). Music Perception, 21, 159-216.

Bellgard, M. I. & Tsang, C. P. (1994). Harmonizing music the Boltzmann way. Connection Science, 6, 281-297.

Bharucha, J. J. (1984). Event hierarchies, tonal hierarchies, and assimilation: a reply to Deutsch and Dowling. Journal of Experimental Psychology: General, 113, 421-425.

Bharucha, J. J. (1987). Music cognition and perceptual facilitation: A connectionist framework. Music Perception, 5, 1-30.

Carpenter, G. & Grossberg, S. (1987). ART2: self-organization of stable category recognition codes for analog input patterns. Applied Optics, 26, 4919-4930.

Carpenter, G. & Grossberg, S. (1992). Neural networks for vision and image processing. Cambridge, Massachusetts: MIT Press.

Castellano, M. A. & Bharucha, J. J. & Krumhansl, C. L. (1984). Tonal hierarchies in the music of north India. Journal of Experimental Psychology: General, 113, 394-412.

Cuddy, L. L. & Cohen, A. J. & Mewhort, D. J. K. (1981). Perception of structure in short melodic sequences. Journal of Experimental Psychology: Human Perception and Performance, 7, 869-883.

Davey, N. & Adams, R.G. (2002). Stochastic dynamics and high capacity associative memories. Neural Information Processing, ICONIP '02. Proceedings of the 9th International Conference 4, 1666- 1670.

Deutsch, D. (1980). The processing of structures and unstructured tonal sequences. Perception and Psychophysics, 28, 381-389.

Deutsch, D. (1982). Grouping mechanisms in music. In D. Deutsch (Ed.), The Psychology of Music (pp. 99-134), New York: Academic Press.

Dowling, W. J. (1978). Scale and contour: Two components of a theory of memory for melodies. Psychological Review, 85, 341-354.

Dowling, W. J. & Harwood, D. L. (1986). Music cognition. San Diego: Academic Press Inc.

Duda, O. & Hart, P. E. & Stork, D. G. (2001). Pattern classification – Second edition. New York: Wiley-Interscience Publication.

During, J. & Mirabdolbaghi, Z. & Safvat, D. (1991). The art of Persian music. Washington D. C.: Mage Publishers Inc, (pp. 31-37).

Gorodnichy, D. O. (2001). The influence of self-connection on the performance of pseudo-inverse autoassociative networks. Radio Electronics. Computer Science. Control, 2, 49-57.

Griffith, N. (1994). Development of tonal centers and abstract pitch as categorizations of pitch use. Connection Science, 6, 155-175.

Griffith, N. & Todd, P. M. (1994). Musical networks: Parallel distributed perception and performance. Cambridge, Massachusetts: MIT Press.

Grossberg, S. & Govindarajan, K.K. & Wyse, L.L. & Cohen, M.A. (2004). ARTSTREAM: a neural network model of auditory scene analysis and source segregation. Technical Report CAS/CNS TR-2003-017, Boston University. Neural Networks, 17, 511-536.

Hendrik, P. & Blankertz, B. & Obermayer, K. (2000). Computing Auditory Perception. Organised Sound, 5, 159-171.

Hinton, G. E. & Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. In D. E. Rumelhart & J. E. McClelland (Eds.), Parallel Distributed Processing (pp. 282-317). Cambridge, Massachusetts: MIT Press.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. Biophysics – Proc. Natl. Acad. Sci. USA, 79, 2554-2558.

İzmirli, Ö. & Bilgen, S. (1996). A model for tonal context time course calculation from acoustical input. Journal of New Music Research, 25, 276–288.

Judetz, E. P. (1996). Türk mûsıkî kültürünün anlamları (Meanings of Turkish music culture). İstanbul: Pan Yayınları.

Justus, T. C. & Bharucha, J. J. (2002). Music perception and cognition. In S. Yantis & H. Pashler (Eds.), Stevens' Handbook of Experimental Psychology, 1, (pp. 453-492).

Karadeniz, E. K. (1983). Türk mûsıkîsinin nazariye ve esasları (Theory and foundations of Turkish music). İstanbul: Türkiye İş Bankası Kültür Yayınları.

Kohonen, T. (1978). Associative memory: a system-theoretical approach. New York: Springer-Verlag, Berlin Heidelberg.

Krumhansl, C. (1990). Cognitive foundations of musical pitch. New York: Oxford University Press.

Krumhansl, C. L. (2004). The cognition of tonality – as we know it today. Journal of New Music Research, 33, 253-268.

Krumhansl, C. & Shepard, R. N. (1979). Quantification of the hierarchy of tonal functions within a diatonic context. Journal of Experimental Psychology: Human Perception and Performance, 5, 579-594.

Krumhansl, C. L. & Toiviainen, P. (2003). Measuring and modeling real-time responses to music: the dynamics of tonality induction. Perception, 32, 741-766.

Little, W. A. (1974). The existence of the persistent states in the brain. Mathematical Biosciences, 19, 101-120.

Longuet-Higgins, H. C. & Steedman, M. J. (1971). On interpreting Bach. Machine Intelligence, 6, pp.221-241.

MacKay, D. J. C. (2003). Information theory, inference, and learning algorithms. Cambridge: Cambridge University Press.

Mitchell, T. (1997). Machine learning. New York: McGraw-Hill International Editions.

Neto, R. C. & Fontanari, J. F. (1998). Categorization in the pseudo-inverse neural network. Journal of Physics A, 31, 531-540.

Norman, K. A. & O'Reilly, R. C. (2003). Modeling hippocampal and neocortical contributions to recognition memory: A complementary-learning-systems approach. Psychological Review, 110, 611-646.

Page, M. P. A. (1994). Modelling the perception of musical sequences with self-organizing neural networks. Connection Science, 6, 155-175.

Özkan, İ. H. (1984). Türk mûsıkîsi nazariyatı ve usûlleri (Turkish music theory and rhythms). İstanbul: Ötüken Yayınları.

Popper, K. R. (1959). The logic of scientific discovery. Hutchinson, London (pp. 248-281).

Segura, E. C. (2001). Associative memories: randomness, continuity and biological plausibility. Proceedings of CACIC 2001, El Calafate, Argentina.

Tanrıkorur, C. (2003a). Müzik, kültür, dil. (Music, culture, language). İstanbul: Dergâh Yayınları.

Tanrıkorur, C. (2003b). Osmanlı dönemi Türk mûsıkîsi (Ottoman era Turkish music). İstanbul: Dergâh Yayınları.

Temperley, D. (2001). The cognition of basic musical structures. Cambridge, Massachusetts: The MIT Press.

Temperley, D. (2004). Bayesian models of musical structure and cognition. Musicae Scientiae, 8, 175-205.

Tillmann, B. & Bharucha, J. J. & Bigand, E. (2000). Implicit learning of tonality: A self-organizing approach. Psychological Review, 107, 885-913.

Todd, P. M. (1989). A connectionist approach to algorithmic composition. Computer Music Journal, 13, 27-43.

Trehub, S. E. & Trainor, L. J. (1993). Listening strategies in infancy: The roots of language and musical development. In S. McAdams & E. Bigand (Eds.), Cognitive aspects of human audition, (pp. 278-327) New York: Oxford University Press.

Vos, P. G. & Van Geenen, E. W. (1996). A parallel-processing key-finding model. Music Perception, 14, 185-224.

Yılmaz, Z. (2001). Türk mûsıkîsi dersleri (Lessons on Turkish music). İstanbul: Çağlar Yayınları.

APPENDIX

COMPLETE PITCH NAMES