Technical Report

UCAM-CL-TR-472
ISSN 1476-2986

Number 472

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

Simulating music learning with autonomous listening agents: entropy, ambiguity and context

Ben Y. Reis

September 1999

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

https://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted July 1999 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Queens' College.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

https://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986
DOI *https://doi.org/10.48456/tr-472*

Abstract

Music learning describes the gradual process of acculturation through which listeners in different cultures develop diverse sets of musical preferences and intuitions. This dissertation describes *Maestro*, a system designed over the course of this research to simulate certain aspects of music listening and learning.

In order to maintain the unbiased flexibility necessary for handling music from different styles, Maestro does not incorporate any *a priori* style-specific knowledge into its design. Instead, Maestro is based on a bottom-up approach that maximises the use of the perceptual information present in a performance.

Maestro's operation involves four stages: it first segments a musical performance on-line according to perceptual cues (segmentation), and constructs an appropriate model of the performance (modelling), based on the context modelling paradigm. This model is simultaneously used to generate expectations about upcoming events (prediction) and to interpret events once they have arrived (parsing).

Ambiguity is an essential aspect of music listening, especially in the context of learning, and can cause multiple hypotheses of interpretation to arise. A novel multi-agent methodology is developed and incorporated into Maestro for generating, maintaining, and reconciling these hypotheses. An information theoretic approach, based on measuring two types of entropy, is used to objectively evaluate the system's relative prediction performance. It is also found that entropy, along with a measure of agent activation, is useful for identifying and classifying different types of ambiguity.

Experiments performed with a collection of 100 Bach chorale melodies provide a basis for comparison with previous machine modelling research and with data from human subjects. A much larger collection of roughly 8,000 folk songs from different cultures enables significant large-scale and pan-stylistic music learning experiments to be performed. Perceptually guided segmentation is argued to yield more cognitively realistic context models than other methods, and is also empirically shown to yield more efficient models for prediction. Additionally, an adaptive modelling strategy allows appropriate multiple-step-ahead predictions to be generated. Finally, a distributed, agent-based parsing methodology is developed and implemented.

The system provides insights into what implications certain theories from cognitive musicology have when put into practice. Maestro's flexible design together with the range of experiments performed and the diverse corpus of musical data enable a thorough and systematic machine-simulated study of key aspects of music learning to be carried out.

Acknowledgements

This work owes a great deal to a number of people.

William Clocksin of the Cambridge University Computer Laboratory provided guidance and supervision throughout my time at Cambridge. Ian Cross of the Cambridge Faculty of Music advised on all matters musical. Both reviewed the manuscript.

This work owes a great debt to the late Professor Helmut Schaffrath of Essen University, whose life work assembling large corpora of folk songs is still enabling research in computational musicology to be carried out. Don Anthony and Eleanor Selfridge-Field at CCARH, Stanford University helped to provide this much needed musical data, making this research possible. Darrell Conklin provided data enabling the present research to be compared quantitatively with earlier work.

This research benefitted from discussions along the way with Emilios Cambouropoulos, Sandra Trehub and Gerhard Widmer, and correspondences with David Cope, Yuzuru Hiraga and Ian Witten. Benjamin Ellis, Miki Grahame, Rivka Isaacson, Boaz Lerner and Fabien Petitcolas gave constructive comments on the manuscript.

The first two years of this research were funded by The Marshall Aid Commemoration Commission. The final year was sponsored by the Cambridge Overseas Trust Marshall Commission Cambridge Scholarship. Thanks to Catherine Reive at the Marshall office for the friendly help and assistance throughout the three years.

Thanks to Cambridge University and the Computer Laboratory for providing an atmosphere conducive to productive research. Thanks to the gang at the computer lab, particularly Mark Humphrys, Kona McPhee, Ian Lewis, Mantej Dhatt, and the members of the Science and Music group.

Thanks to my officemate Boaz Lerner, who was always ready to listen, help and advise.

Thanks to my friends at 3 Thompsons Lane, David for being there on the phone Friday mornings, and Elliot and Jonny for making sure a sound mind remained in a sound body.

Thanks to my sisters Tal, Chen, and Mor for their help along the years. Finally, to my parents, who brought me up to appreciate those things in life that are most important. This dissertation is dedicated to them.

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration. This dissertation is not substantially the same as any I have submitted for a degree or diploma or any other qualification at any other University. No part of this dissertation has already been, or is being currently submitted for any such degree, diploma or other qualification. The text has fewer than sixty thousand words.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> *When one day an arctic traveller played a recorded song by one of the most famous European composers ... to an Eskimo singer, the man smiled somewhat haughtily and stated, "many many notes, but no better music" (Sachs, 1965).*

People are affected by their previous musical experiences [68, 115, 127]. Consider the case of two identical twin brothers separated at birth – one grows up in Beijing, the other in Berlin. Despite their initial similarities, the twins develop significantly different sets of musical preferences and intuitions as a result of the diverse musical cultures in which they grow up. These differences might express themselves in the performance of certain musical tasks, such as completing a partially-played tune or determining the structure of a tune which is heard. Interesting experiments can be performed to examine how musical experiences during childhood can shape the twins' different intuitions and preferences at maturity.

Consider further what would happen if at a certain point during development the twin living in China moves to Berlin. To what extent will he remain biased from his earlier experiences? To what extent will he incorporate the new musical surroundings into his intuitions? How does the degree of similarity between the two musical styles affect these phenomena?

This dissertation describes research aimed at addressing these questions and other related issues. A functional machine model of certain aspects of music cognition and learning is developed. The model is designed according to cognitive principles and is used to perform large-scale experiments in music learning.

This introductory chapter will begin by formulating the problem to be studied and the approach taken. It will then introduce *Maestro*, the machine model developed over the course of this research to conduct a simulated experimental study of music learning.

1.1 Objective: Studying Music Learning

As people listen to pieces from a particular style, they acquire a set of preferences and intuitions inherent to that style. Lerdahl and Jackendoff [72] describe *musical intuitions* as the largely unconscious knowledge that a listener uses in listening to a piece of music. This knowledge enables the listener to organise and make sense of the patterns present in the music, to identify elements of a piece as typical or anomalous, to recognize various kinds of structural repetitions and variations, and generally to comprehend a piece within the style.

Music learning is defined here as the process of developing a set of musical intuitions and preferences characteristic of a certain musical style, based on listening to examples of that style. This learning enhances a listener's abilities to organise music into appropriate structures and to generate style-pertinent expectations when hearing passages of music from a particular style [127, p. 58]. Sloboda divides music learning into two categories. *Enculturation* consists of a shared set of experiences which the culture provides as children grow up. *Training* refers to formal instruction [115, p. 196]. While training is an important aspect of musical development [84, 116], this research focuses specifically on the effects of enculturation.

The phenomenon of music learning is closely associated with the notion of musical style. Therefore, before proceeding to explore music learning, it is appropriate to clarify what is meant by musical style, and how different types of learning relate to it.

1.1.1 Levels of Learning

Narmour identifies two levels of musical style [86]. *Intra-opus style* refers to the commonalities shared by different parts of the same musical piece – for example, the thematic motifs reappearing across the various movements in a Beethoven symphony. Conversely, *extra-opus style* refers to the commonalities shared by different pieces originating from the same composer, group of composers, culture, geographical area, or time-period [80]. For example, certain cadential patterns might be used repeatedly by Beethoven in many of his different works, while looser, wider-scope commonalities may be found between various pieces of Western Renaissance music, or even Western music in its entirety.

Music learning can be said to take place on these two levels, both within an individual piece, and, on a larger scale, over an entire musical style. When hearing a new piece of music, a person must gradually learn to listen to it. The more a person is exposed to that specific piece, the more he learns to hear and discern, and thus the more he can detect and appreciate in the music. This occurs with music that is new to the person, and continues

even once the music becomes more familiar. This process is called *intra-opus learning.* A person must also gradually learn to listen to and recognise the stylistic invariants appearing across pieces of music from that style. By listening to many sample pieces over time, the person internalises the intuitions of the style. This process is called *extra-opus learning.*

Jackendoff [61] states that one of the eventual goals of research in musical cognition is to understand how a novice comes to be able to understand music. Rosenthal takes this point even further and states that the issue of music learning constitutes the "profound question" of music cognition research [98]. Given this ample motivation for studying music learning, the objective of this research is to study issues in music learning by performing experiments with music from different styles.

1.2 Research Approach: Machine Modelling

1.2.1 Difficulties with Human Experimentation

Conducting music learning experiments with human subjects poses two main problems. First, a person's collected life-time musical experiences and the resulting internalised intuitions are crucial factors in determining the outcome of music learning and listening. Widmer [127] refers to this as the individual listening history of a human listener, with all its related implications. These factors can be highly complex and would be extremely difficult to monitor accurately in human subjects, let alone control in the systematic way necessary for proper experimentation.

The second problem involves studying the moment-by-moment perceptions of a listener. This is critical for attaining a deeper understanding of the processes involved in music learning. Hiraga calls this *incremental processing*, and explains that musical processing proceeds with the flow of music, deriving results at every instant [55]. However, it is virtually impossible to obtain a step-by-step record of a human listener's ongoing perceptions in a realistic musical context. Sloboda [115] explains that since the end-product of listening to music is a set of fleeting, uncommunicable mental images, the psychologist is at a considerable disadvantage when trying to tap the moment-by-moment history of mental involvement with the music. Sloboda concludes that this is the principal problem facing researchers in music listening. Similarly, Krumhansl [68] points out that various experimental methods for studying musical expectancy necessarily interrupt the musical experience, and so generalisations to more typical listening conditions must be made with caution. Berent and Perfetti [8] have similar reservations about any experimental technique that necessarily terminates the listening process.

Sloboda notes an additional complication. Experiments are usually performed with only short selections of music. This methodology may carry with it certain implicit assumptions, and Sloboda cautions against extrapolating the findings of these experiments to longer, more realistic musical contexts. With regard to performing experiments to study listening during longer musical contexts, Sloboda comments that there is no satisfactory way of tapping what goes on when music of more extended proportions is heard [115, p. 191].

1.2.2 Machine Modelling

Machine modelling addresses the main difficulties of human experimentation. First, total and exact control can be exercised over the prior musical experiences of the system. Second, continuous monitoring of the system's performance and perceptions throughout the experiments can be easily achieved. Conklin and Witten [132] point out that when using a computational model it is possible to examine in detail how the model views particular musical events by inspecting its inner workings at each stage of computation. Additionally, Minsky [82] proposes that machine simulation opens up opportunities that other, more formal analytic approaches do not. He suggests that future machine simulations might raise simulated infants in traditional musical cultures.

With the benefits inherent to the machine modelling approach, however, come the significant challenges of modelling a music learning system.

1.2.3 Challenges: Learning by Listening

Music learning is not a holistic phenomenon whose results can only be detected after learning is complete. Rather, it is a gradual process. Learning takes place over an extended period of time, with incremental changes occurring with each listening. As Meyer points out, musical expectations are affected by one's knowledge of the musical style, while at the same time new musical experiences alter one's expectations about the style as a whole. As a result, the "internalised probability system" [61] is constantly undergoing subtle change, and one's expectations differ, even if slightly, on successive rehearings of a piece. Music learning happens as a result of music listening, and any attempt to model music learning by definition must rely on a model of music listening.

The learning by listening approach has important implications for the design of a machine model. Namely, the processes involved in learning, such as storing information from previous experiences, should result directly from the processes involved in music listening, such as segmenting and interpreting the performance.

Modelling music listening is a complex and challenging problem in its own right. It involves the real-time processing of parallel, temporally ordered event sequences, and the identification of significant patterns present at many levels in the data. The processing of the data is heavily dependent on the context of the current piece as well as on the listener's past musical experiences. Musical data itself is informationally rich, and is multidimensional, including information on pitch, time, loudness, as well as other parameters. Much is unknown about the immensely complex nature of music cognition, and, as such, a complete cognitively realistic model of music learning and listening is an obviously premature goal [72, p. 8].

1.2.4 Design Principles

Since a full model of music cognition is not realistic, this research adopts a "functionalist" approach [61] that focuses on certain aspects of particular relevance or interest. First, this research focuses on those aspects of music cognition that are particularly relevant to music learning and thus necessary for performing the experiments described above. These include maintaining the flexibility to handle music from different styles and making analysis decisions based on bottom-up perceptual cues rather than pre-encoded knowledge. It also includes explicitly handling musical ambiguity which is especially important in the context of learning. With regard to such a focused approach, Rosenthal comments that machine models of music cognition, while they are necessarily limited and rest on many assumptions, they can nonetheless be implemented in sufficient detail to provide fresh and interesting insights into the problem [98].

A second criterion is added in the present research in order to reduce the number of limitations and assumptions referred to by Rosenthal: cognitive realism. Previous research in machine learning of musical styles has focused on achieving optimal results – be it for music prediction [132], style discrimination [62, 125], music performance [73, 129], or composition in a certain style [31, 91, 126]. In the present research, while results are measured by gauging prediction performance, the goal is not achieving optimal results. Rather, the focus is on developing and studying a *more cognitively realistic model* of the music learning process.

To achieve this, the design of the system's learning capability is guided by principles derived from cognitive musicology and experimental psychology. Perceptual cues are used to segment the music, and constraints are placed on model growth rate and memory usage. Additionally, the data sets used for training are of a significantly larger, more realistic order of magnitude. These guidelines also serve to make the results produced by the model more relevant to human music cognition.

These two design principles are elaborated into a detailed set of design specifications in Chapter 2. With the objective of the research formulated

and the approach to the problem laid out, the stage is now set for introducing Maestro, the system designed to carry out this research.

1.3 Maestro

This dissertation describes Maestro[1], a system developed over the course of this research to simulate certain aspects of music listening and learning. Maestro is designed primarily for performing large-scale music learning experiments, described below in section 1.3.5. Since music learning is inextricably tied with music listening, Maestro's design also includes a model of music listening with a learning focus. Maestro's design is based on two key principles: cognitive realism and a focus on flexibility and learning.

1.3.1 System Operation

Maestro is a program containing approximately 10,700 lines of C++ code, running on a Pentium-based Linux system. The Tcl/Tk toolkit [88] is used for graphically displaying the system's output, as specified in about 1,100 lines of custom Tcl/Tk scripting code. Two additional utilities written to assist the research are described in Appendix C.

Maestro has four main stages of operation (Figure 1.1): Segmentation, Modelling, Prediction, and Parsing. Each is performed in turn when Maestro processes a musical event. A detailed description of Maestro's event-loop can be found in Appendix A.

Maestro takes in music in a form derived from MIDI that includes pitch information in semi-tones and onset times and durations (not necessarily quantised into standard note-lengths). This can be interpreted as being similar to a reduced form of the *musical surface* mentioned by Jackendoff [60]. Maestro processes the music on-line, one note-event at a time.

A fresh instantiation of Maestro begins with no musical experience in its knowledge base. Instead, it continually learns as it listens, building up an internal model of the knowledge gained from its musical experiences. This learning process involves two steps:

- **Segmentation**: Perceptual cues in the music are used to indicate possible ways to break the music into salient chunks for storage in memory.
- **Modelling**: A model that captures both the structure and content of the musical data is constructed according to the various segmentation possibilities.

[1]The name **Maestro** derives from the common acronym for **M**ulti **A**gent **S**ystem.

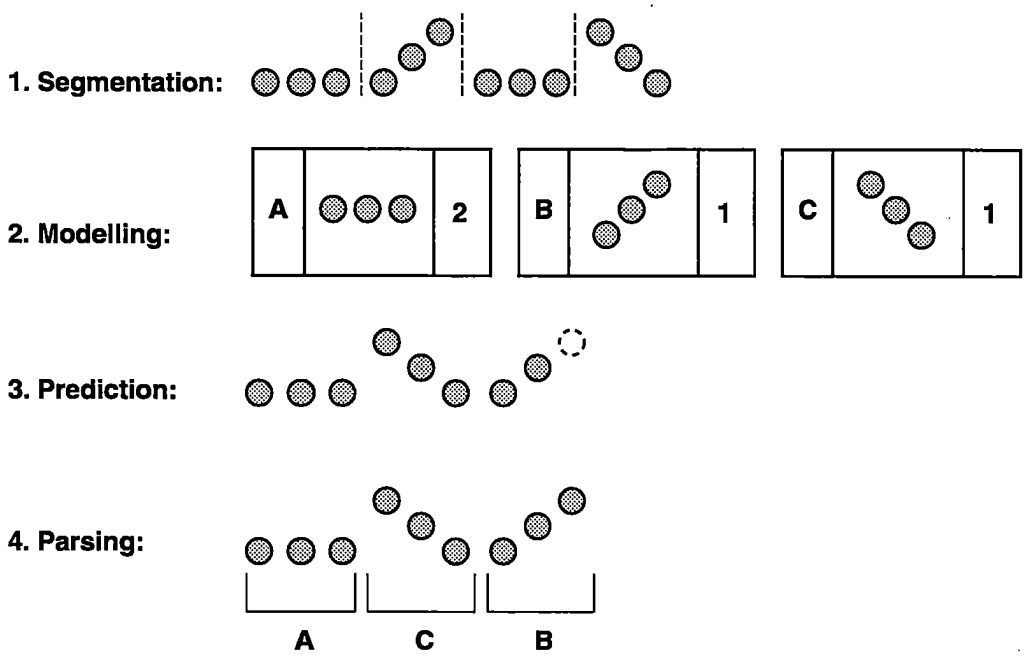


Figure 1.1: The four stages of Maestro's operation: The input is segmented (1) and the resulting segments are stored into a model (2). This model is then used to predict future musical input (3) and to label repetitions of previously stored patterns (4).

As Maestro learns with experience, it simultaneously uses its developing model to perform the following two musical tasks:

- **Prediction**: Appropriate multiple-step-ahead predictions are generated concerning what is to come next in the music.

- **Parsing**: The structure of the music is labelled according to the interpretation arrived at using the model.

The learning process continues throughout listening as Maestro's internal model is constantly being updated. This is noticeable in the improving prediction performance of the system.

By way of introduction, three key technical aspects of Maestro's design are now briefly presented: The modelling methodology, the use of agents and the performance measures employed.

1.3.2 Methodology: Context Models

Maestro stores musical information as segments of pitch intervals in a *context model*, which keeps track of segments of pitches encountered in the past for the purpose of predicting pitches in the future. Context models have been used previously for studying music composition [14, 91] and music prediction [27], and are formally introduced during the discussion of Maestro's modelling stage.

Maestro's context model is designed to be more cognitively realistic than its predecessors. More realistic constraints are placed on model size and growth. Additionally, the model is capable of storing contexts of variable lengths, providing it with increased flexibility, making it more efficient, and allowing it to generate appropriate multiple-step-ahead predictions.

1.3.3 Activated Modelling: Listening Agents

Maestro incorporates the *multi-agent system* paradigm into its design. An *agent* is an autonomous entity able to take certain actions to accomplish a set of goals. A *multi-agent system* is a collection of independent agents, each working toward its own goals. The agents in such a system might operate alone, interact with each other, cooperate, compete, and can also learn. From the actions and interactions of the individual agents, the complex behaviour of the system as a whole emerges [40].

Unlike previous systems [102, 132], Maestro does not use its context model to passively generate predictions. Instead, the context model is activated through the instantiation of *active listening agents* that serve to represent and advocate different possible hypotheses of musical interpretation.

The incorporation of the multi-agent system paradigm into Maestro's design serves two goals: First, this research follows the active listening approach – the view held by Reiss Jones [64], Minsky [82], and Rosenthal [98], among others, that musical memory and musical processing are inextricably connected. Through the use of agents, memory and processing are merged into one. Second, this research follows the multiple hypothesis approach – Jackendoff's [61] theory that musical ambiguity is handled by maintaining multiple hypotheses of interpretation. Agents are used to represent various hypotheses of interpretation.

In Maestro, the prediction and parsing tasks mentioned above are actually performed by the individual listening agents in a distributed fashion. Agents can generate conflicting predictions which must be integrated to form the system's prediction as a whole. A distributed agent-based parsing algorithm is developed to handle parsing ambiguity. The agents compete and cooperate with each other in performing these tasks, and the desired

system-wide behaviour emerges from the interactions of the independent agents.

The implementation of Maestro's active listening agents takes advantage of the object-oriented capabilities of C++. See Appendix B for a detailed description of the C++ class declaration for Maestro's listening agents.

1.3.4 Performance Measure: Entropy

In order for any experiments in music learning to be conducted, an appropriate measure of performance is needed to monitor the progress of the learning. In the particularly complex context of music, careful thought must be given to choosing an appropriate performance measure. Three options are considered:

The first option is the *generative approach*, which involves asking the listener (in this case the system) to compose a new piece in the style of what has already been heard. However, as pieces within the same style can often differ significantly from one another, the evaluation of the stylistic adherence of a certain piece can prove difficult [125], if not somewhat subjective. A more objective measure is needed.

Another approach, used by Westhead and Smaill [125], is the *recognition of works in a style*. The listener is trained with music from at least two different styles and is then asked to classify a new piece as belonging to a particular style. This approach requires presenting the listener with songs from different styles every time a measurement of music learning is to be taken, and makes difficult the continual tracking of learning during listening. Furthermore, the purpose of the present research is to study the learning of a musical style in its own right, and not only in relation to other styles. As a result of these difficulties, this method is not chosen here. Still, the issue of discriminating between music from different styles is addressed by the present research in the multi-style experiments reported in this dissertation.

A third approach for measuring music learning is the *prediction method*, or the *sung continuation procedure*: after hearing only part of a melody, the listener is asked to predict what will come next. Adachi and Carlsen [1] cite the view that this experimental method is best for reproducing the spontaneity of the anticipation processes during actual music listening. Since the prediction method is an objective measure that also allows for continual measurement of musical learning, it is the method of choice in this research.

Conklin and Witten [132] measure prediction performance using an information theoretic approach that involves calculating a type of *entropy*. In Maestro, two types of entropy are used, and experiments are performed to explore various types of music learning from different perspectives. The two types of entropy are also used to classify and study different types of ambiguity experienced during listening.

1.3.5 Experiments

Maestro is developed to conduct four basic types of experiments to study music learning: Style Learning, Style Switching, Comparative Listening and Geographical Mapping.

1. Style Learning experiments involve presenting the listener with many examples of one style of music (e.g., the works of Bach). One can then monitor the increasing understanding of the musical style by seeing how well the listener predicts as-yet-unheard samples of the style. The prediction should improve with additional listenings.

2. After being trained in one style of music, it is interesting to observe a listener's reaction to a totally different style of music, as well as how the listener proceeds to develop an understanding of the new style. Such Style Switching experiments address some of the issues introduced in the "twins" example above.

3. Comparative Listening involves presenting the same piece of music to two listeners from different musical backgrounds and comparing their relative levels of prediction performance. This also provides information regarding the way in which prior musical experiences affect music listening. One would expect a native listener to outperform a foreigner when listening to a native piece, as reported by Lynch *et al.* [75]. Similarly, Zielinska and Miklaszewski report that "culturally remote" melodies are more difficult to memorise [133]. However, Blacking [18] notes that a foreigner would still be able to make *some* sense of the music without having to spend much time learning the new style. The present research studies the extent to which experience through simple exposure to certain musical styles prepares a listener for dealing with music from other musical styles.

4. The final type of experiment is Geographical Mapping. Sloboda [115] reports on experiments in which children were asked to judge whether two musical selections were from the same piece of music. The results showed a greater ability to correctly differentiate 'different' pairs when they came from widely different stylistic eras. For instance, subjects were better at discriminating Boulez from Bach, with 300 years between them, than Schumann from Brahms, who were contemporaries. It is also interesting to perform experiments where the musical styles are separated geographically instead of chronologically. Figure 1.2, taken from [25], shows one hypothetical relationship between geographical musical influence and musical stylistic similarity. These relationships result from the musical influences of interacting cultures. The Geographical Mapping experiments reported in this dissertation

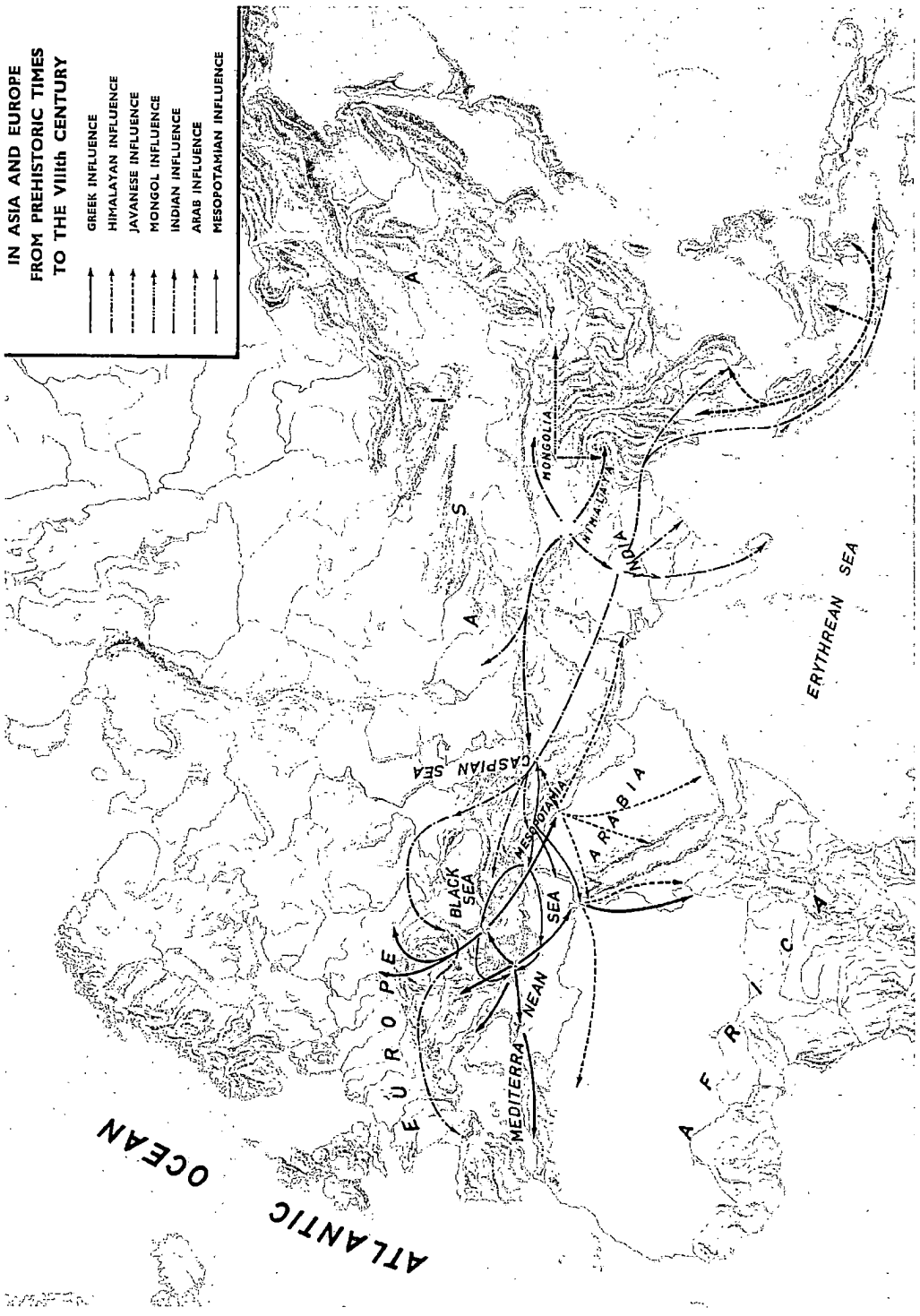


Figure 1.2: Map showing different geographical musical influences. From (Collaer and Linden, [25]).

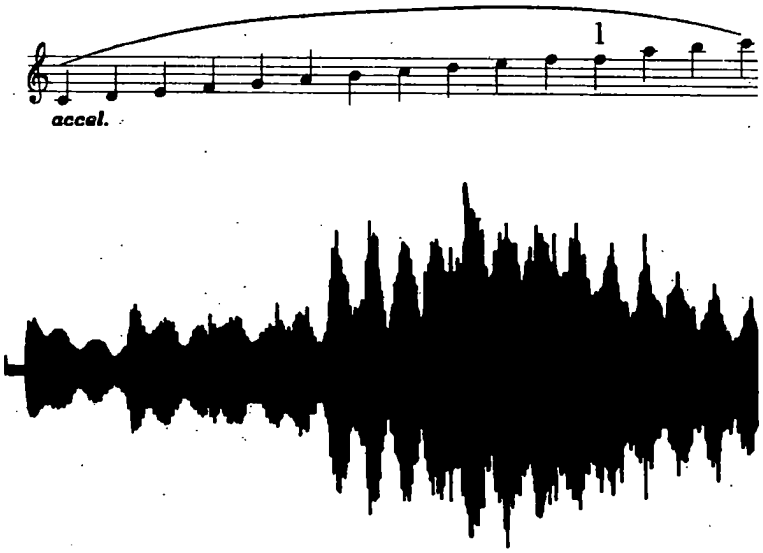


Figure 1.3: Audio correlates of musical notation in the time domain. From (Foster *et al.*, [46]).

attempt to identify correlations between musical similarity and geographical proximity of different musical styles.

1.4 Research Scope

Before reporting the work undertaken in the rest of this dissertation, it is important to define clearly the topics addressed by this research, as well as those that are not. Maestro is by no means a complete model of human music cognition. Rather, it represents an attempt at modelling certain aspects of music listening and learning in a way that is more cognitively realistic than previous systems.

1.4.1 Focus on Learned Component

Musical intuitions consist of both learned and innate components [18, 68]. Many theorists point out that the innate component forms the basis for the universal musical invariants found across all cultures [72, 115]. Based on literature from experimental psychology [69] and music theory [86], it is unrealistic to assume that an infant begins life with absolutely no innate

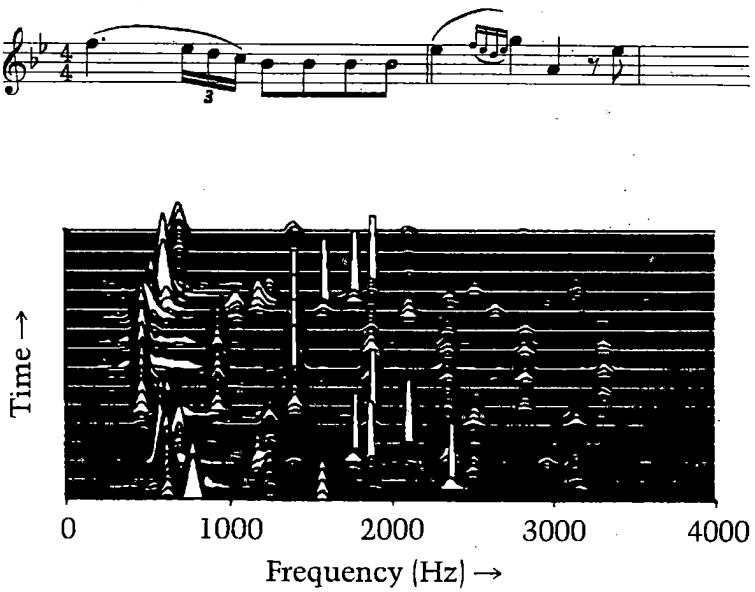


Figure 1.4: Audio correlates of musical notation in the frequency domain. From (Foster *et al.*, [46]).

musical preferences. However, Maestro does not model people's innate musical preferences, and this exclusion of innate knowledge allows this research to focus solely on the learned component of music intuition – namely, those effects resulting from experience.

1.4.2 Abstraction Away from Audio

Low-level audio aspects of music, such as timbre, attack and envelope are essential to modelling the complete music cognition experience. A large amount of complex processing goes on at this level that has certain crucial influences on the final perception of the music (e.g., Figures 1.3 and 1.4). These effects are addressed by research in auditory physiology [13, 44] and machine modelling of music cognition [16, 122, 123].

Low-level audio features are not included in the data representation used by Maestro and this abstraction is an important limitation of the present research. However, the field of audio processing is in itself so complex that its inclusion would expand the scope of this project beyond a reasonable size. Widmer suggests that a level of abstraction should be found which combines transparency and simplicity with at least a minimum of musical and psychological plausibility [127]. The approach taken here is that the musical

information encoded in the "musical surface" format (see Appendix C) is indeed sufficient to allow a relevant and effective study of music learning.

1.4.3 Monophony

Maestro deals only with monophonic melodies. Most music in the world today is polyphonic, and takes advantage of the extra dimensionality polyphony provides. However, the introduction of polyphony greatly increases the complexity of the music listening and learning problem. The results reported in this dissertation show that monophonic input, especially with large data sets, is sufficient for conducting informative and meaningful experiments in music learning.

1.4.4 No Explicit Tonality

Tonality plays a central role in the musical experiences of most world cultures. Many researchers, including Cross [35, 38, 58] and Krumhansl [67] have focused their efforts on studying pitch schemata, which can reach great complexity and variety in different cultures [44, 121]. However, tonality is for the most part style-specific, and an explicit representation of a system of scales and chords would likely limit Maestro's ability to handle music from different styles. Therefore, this lies outside the scope of this research.

1.4.5 Limited Handling of Rhythm and Timing

Generating rhythm and timing expectations is a crucial aspect of human music listening and much work has been done in computational modelling of human rhythmic perception [23, 99]. Maestro makes use of timing information in its segmentation stage, as described in Chapter 3. However, Maestro monitors, predicts and parses patterns of pitch only, and does not store any timing information. This approach is consistent with Narmour's theoretical model which states that pitch expectations are generated solely from previous pitch information and do not rely on timing information [68, p. 60].

1.4.6 Exact Pattern Matching

Music cognition involves the ability to handle inexact pattern matches. Regarding the question of what constitutes a match, Hiraga [56] says that there seem to be no definitive criteria, but that these may include persistence of the pattern, strength of the match, temporal proximity and complexity of the pattern itself. Overill [89] also discusses issues relating to inexact pattern matching and describes various thresholds that need to be set to determine the tolerance of the matching.

The introduction of inexact pattern matching would, apart from adding further complexity, make the objective prediction measurements used in this research dependent on whatever arbitrary matching thresholds were chosen. This would allow too much variability in measuring performance. Exact matching, also used by Westhead and Smaill [125], proves sufficient for the systematic and controlled study of music learning pursued in this research.

Having given a brief overview of system operation and project scope, this chapter now concludes by highlighting the main contributions of the present research and presenting a brief outline of the rest of the dissertation.

1.5 Objectives and Contributions

This research serves several objectives. First, it enables the simulated experimental study of music learning by developing a new model of music cognition that is in certain ways more cognitively realistic than previous research. As described above, attempting to take similar measurements and to perform similar experiments with human subjects would prove extremely difficult, if not impossible. A machine modelling approach is thus chosen.

Second, this research tests certain theoretical models of music cognition by implementing them in a computer simulation. These include Minsky's agent-based music cognition paradigm [82] and Jackendoff's parallel multiple analysis model of on-line musical parsing [61].

Third, the research novelly combines context-model-based music learning methods [14, 91, 132] with an agent-based simulation of music cognition [55, 98, 103], both emerging areas of research in computational musicology in which significant work remains to be done.

The following are the main contributions of the research presented in this dissertation, presented in three groups: features of Maestro's design, experiments performed with Maestro and tools developed for analysis of the experimental results.

1.5.1 Design Features

- A Clean Slate – Maestro is a machine model of music listening and learning that incorporates no *a priori* style-specific knowledge in its design. This endows it with a flexibility to learn music from many different styles.

- Perceptually Guided Segmentation – Previous systems have segmented music without regard for bottom-up perceptual cues or the segmentation ambiguity that these cues entail. Maestro's segmentation stage considers three perceptual cues and explicitly handles segmentation ambiguity.

- Adaptive-Order Context Modelling – Previous research with context-model-based systems has ignored certain cognitive constraints relating to model construction and growth rate. A more cognitively realistic context modelling paradigm is devised for music learning. The model contains segments of variable order, each appropriate for the specific musical context.

- Agent-Based Prediction – An agent-based methodology for generating and integrating various predictions is developed.

- Agent-Based Parsing – A distributed implementation of bottom-up, left-to-right, optimal, partial chart parsing is developed. The desired parsing behaviour emerges from the interactions of the individual autonomous listening agents, each suggesting its own parsing interpretation.

- Handling of The Circularity Problem – Music can be grouped in two ways – by searching for points of discontinuity, or by detecting repetition or parallelism. Maestro's segmentation stage handles the first, while its parsing stage addresses the second. The inherent interdependence between these two methods of grouping leads to the *circularity problem* mentioned by Hiraga [56] and Larson [70]. As described in detail in the description of the parsing stage, Maestro uses a *directed search* for patterns to address this issue.

1.5.2 Experiments

- Experiments in Music Learning – Previous systems studying music learning have used relatively small data sets, and have only studied the final state of a trained model. In the present research, large datasets allow for extensive controlled experimental study of the *actual process* of music learning using a real-time, on-line listening and learning system.

- Multiple-Style Experiments – Style Switching and Comparative Listening experiments are performed using samples from many different musical styles.

- Geographical Mapping – Experiments are performed to identify correlations between musical similarity and geographical proximity.

- Multiple-Step-Ahead Prediction – Previous systems have studied predictions generated only one note ahead of time. Maestro produces appropriate multiple-step-ahead predictions on-line, based on its variable-

order context model. A systematic experimental study of the prediction performance is conducted for different forecast horizons and for various levels of training.

- Perceptually Guided Segmentation – It is hypothesised that certain points of segmentation are more meaningful and that a segmentation strategy guided by perceptual principles results in models that are more efficient for the purposes of prediction than other segmentation strategies. Although hinted at in the literature, these hypotheses have not been empirically tested. An experimental method is developed in this research in order to validate this hypothesis.

1.5.3 Analysis Tools

- A Three-Fold Framework for Analysing Music Learning Experiments – A three-fold analysis methodology that measures context model growth, number of predictions generated, and prediction performance is devised and used to analyse the results of large-scale music learning experiments.

- Entropy-Based Ambiguity Classification – Two types of ambiguity are identified: ambivalence and uncertainty. An analysis methodology incorporating two types of entropy and a measure of agent activation is developed to study and classify ambiguity into these two types.

- Entropy-Based Training Measure – A dual-entropy characteristic is developed to gauge the level of experience, or *maturity* of a model.

1.6 Dissertation Overview

Following this introductory chapter, Chapter 2 elaborates on Maestro's design principles of cognitive realism and a focus on learning and derives a complete list of specifications, highlighting the important features of Maestro's design.

The next four chapters address the various stages of Maestro's design in detail. Each of these chapters consists of a review of the relevant cognitive foundations and of previous work in machine modelling, followed by a description of Maestro's implementation. Chapter 3 covers the design of Maestro's segmentation stage, comparing different segmentation strategies and addressing issues of segmentation ambiguity. Chapter 4 considers different modelling strategies, memory types and storage formats for Maestro's modelling stage. Context models, the active modelling approach and the implementation of listening agents are all described in detail. Chapter 5 deals with Maestro's prediction capabilities, including prediction ambiguity

and multiple-step-ahead predictions. Entropy-based performance measures are formally introduced. Chapter 6 presents Maestro's approach to parsing, as well as the distributed agent-based parsing algorithm developed to implement that approach, capable of handling parsing ambiguity.

The next four chapters include the main experimental portion of the dissertation. Chapter 7 discusses the results of the various large-scale musical learning experiments. Experiments with 100 Bach chorales used by Conklin and Witten [132] provide a basis for comparison with earlier research and with actual results from human subjects. Much larger data sets containing thousands of songs are then used to study a more complete music learning process. A framework for analysing music learning is developed and used. In Chapter 8, the relationships between different performance measures are analysed, and a method of classifying and identifying two types of ambiguity is developed. A method is also developed to measure the level of training, or maturity, of a model. Using a novel experimental method, N-Note Segmentation Shifting, experiments reported in Chapter 9 provide empirical evidence that Perceptually Guided Segmentation leads to more efficient modelling for the purposes of prediction. Chapter 10 includes further experiments involving music from different styles. Style Switching and Comparative Listening experiments are performed. The full capabilities of Maestro are then displayed in a set of Geographical Mapping experiments, studying geographical relationships between musical styles.

Chapter 11 presents a review of relevant literature and compares Maestro with other related research in the field. Machine models of music learning and ambiguity are reviewed, and agent-based models of music cognition are also presented. Finally, Chapter 12 summarises the main conclusions of the research and highlights a number of opportunities for further work. The Appendices include details of Maestro's event loop, agent class declaration, data representation, utilities developed over the course of the research, and references for the data used in the Geographical Mapping experiments.

Some of the design concepts have been implemented and tested on a prototype system in the early stages of this research, reported in [92]. Portions of this research have also been published in refereed proceedings in [93] and [94], and presented at international conferences.

Chapter 2

Design Features

This chapter introduces the various features of Maestro's design. The two principles of cognitive realism and a focus on learning mentioned in Section 1.2.4 are elaborated into a complete list of design specifications. The motivation for each design specification is described, along with some previous attempts at addressing it, followed by a discussion of the way in which Maestro deals with the issue.

2.1 Flexibility and Learning Focus

Maestro is a system developed primarily for studying certain aspects of music learning. This focus on learning is expressed in three design specifications: pan-stylistic potential, a clean slate, and the use of large data-sets. These three specifications are implemented throughout the various stages of Maestro's design, as described in Chapters 3 through 6.

2.1.1 Pan-stylistic Potential

People from different cultures appreciate and respond to very different styles and systems of music. In a review of ethnomusicology, Titon [121] presents a large assortment of music from different cultures from around the world. Dowling also cites a number of illustrative examples of the extents to which music varies throughout different parts of the world [44]. The wide diversity of musical styles is clearly evident in these selections. Figure 2.1, taken from [25], shows different systems of tonality in various geographical regions.

For each musical style found in the world's cultures, there are people who grow to internalise it into their musical preferences and intuitions. It can therefore be inferred that the musical capabilities of humans are, at least initially, flexible enough to learn the many different types of musical styles.

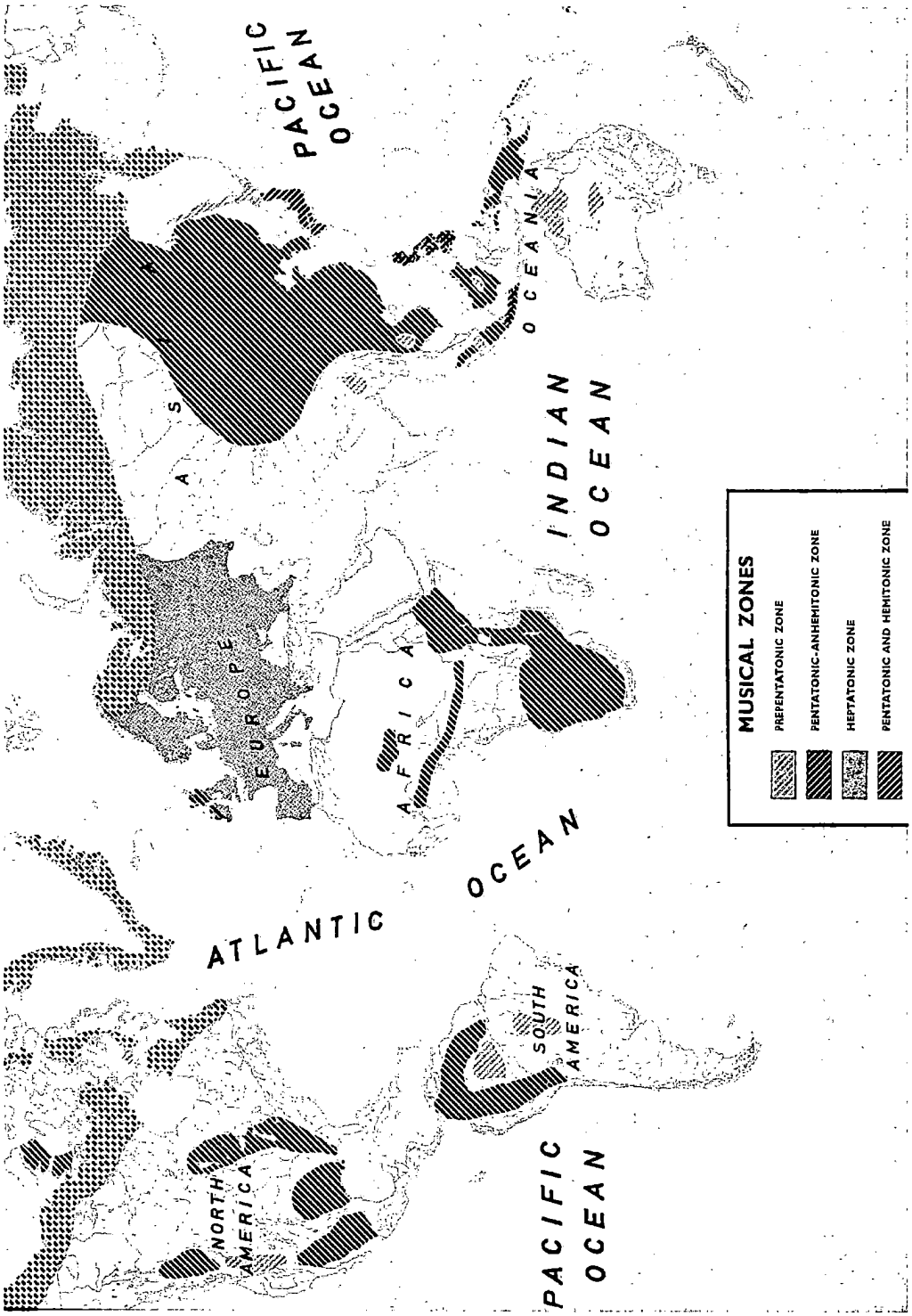


Figure 2.1: Map showing different tonality zones. From (Collaer and Linden, [25]).

In order to emulate the true human musical capabilities, a machine model should be able to develop and learn to perceive different styles of music from all over the world. In line with this approach, Conklin and Witten seek a methodology that is capable of capturing regularities in different musical genres [131]. Their impressive program of research into music learning and prediction did not in the end reach this pan-stylistic goal [27]. Maestro is aimed at addressing this goal.

2.1.2 A Clean Slate

Many music learning systems incorporate certain amounts of stylistic knowledge *a priori*, whether in the form of a knowledge base [127], or as parameters in a model [103]. This knowledge may include such items as common structural forms, tonality systems or chordal structures. Lerdahl and Jackendoff's *Generative Theory of Tonal Music* (GTTM) is described as modelling the intuitions of an "experienced listener" [72, p. 230], and assumes that music learning has already taken place. Similarly, both Rowe [103] and Conklin and Witten [27] pre-program a developed sense of style-specific tonality into their systems.

Endowing a system with style-specific elements *a priori* is known as the knowledge engineering approach. While such information may help in analysing music from a particular style, this optimisation also suffers from three main drawbacks: over-specialisation, incompleteness, and rigidity.

First, the specialisation inherent to the knowledge engineering approach typically limits a system's ability to analyse music from other styles. Significant items may not be processed correctly, and preconceived notions might instead be imposed inappropriately. A second drawback of the knowledge engineering approach, according to Conklin and Witten [27], is that there are too many exceptions to any logical system of musical description. It will therefore be difficult to ensure the completeness of an intuited theory, even for dealing with a single musical style. Third and finally, the knowledge engineering approach by definition ignores modelling the music learning process, which is the prime focus of this research. Flexibility is necessary for learning, and a fixed, pre-set knowledge base is rigid and inflexible.

To model music learning of different styles, a system should not start off with any *style-specific* information that in reality is only learned through experience. Instead, the absence of pre-encoded style-specific information in a system should endow it with the initial flexibility to learn music from many different styles. Hiraga refers to this requirement as *robustness*, and states that the design of a system should not be governed by a presupposition of a particular style or cultural context [55].

Ponsford *et al.* [91] describe a system that maintains its generality by minimising the use of music knowledge. Still, some musical knowledge is included in their system, such as the rules of Western harmony in the prepro-

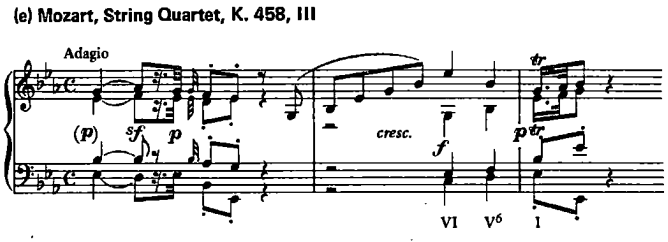


Figure 2.2: Musical notation incorporating key signature, time signature and functional harmony annotations. From (Aldwell and Shachter, [2]).

cessing stage and common phrase structure in the annotation stage. Bryson *et al.* [17] also use little or no musical knowledge, and the function of their system is kept to low-level chord generation. The present research attempts to determine to what degree a system without any *a priori* stylistic information is able to gain proficiency in any given musical style, as measured by its ability to appropriately predict musical events. This general learning approach is maintained throughout Maestro's design and operation.

2.1.3 Large Data Sets

Previous studies in machine modelling of music learning have used medium-sized corpora for training. Brooks *et al.* [14] used 37 hymn tunes, Ponsford *et al.* [91] used 84 French *Sarabandes*, and Conklin and Witten [132] used 100 Bach Chorales.

Since the present research seeks to study a realistic music learning process, it requires a more realistic musical experience base similar to one that would be available to someone growing up in a certain musical culture. A system presented with too few samples of a musical style cannot complete the full learning process: Conklin and Witten mention that performance of their prediction system improved in trials where additional training data was used [132]. To this effect, the experiments performed with Maestro involve much larger data sets containing thousands of songs.

2.2 Cognitive Realism

The second principle guiding Maestro's design is cognitive realism. It is important that a machine model of human music listening and learning be based on some knowledge of how these capabilities operate in humans. First, there is the issue of experimental relevance; if the model is made more cognitively realistic, the results produced can reveal more about how humans

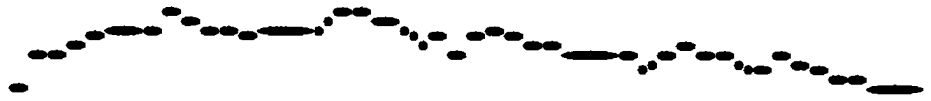


Figure 2.3: The musical surface format used by Maestro, representing pitch in semitones and time not quantised into standard note values. The example is a Bach Chorale melody.

might perform in similar situations. Second, even if the goal were not to study human music cognition, there is still the notion that dealing with music in a vacuum, divorced from its human origins, is meaningless. As Rosenthal points out, music is deliberately tailored to the mind's cognitive limitations and capacities [98]. Therefore, a study of music cognition that ignores critical specifics of human music cognition would not be very meaningful. The design of the system should therefore be guided by evidence from the literature in the fields of experimental psychology and cognitive musicology.

This design principle of cognitive realism is now elaborated into a list of seven design specifications, including:

- Realistic input;
- On-line processing;
- Time and memory constraints;
- Multiple-step-ahead prediction;
- Ambiguity handling;
- Retrospective listening;
- Active listening.

In the ensuing discussions, important features of Maestro's design are highlighted.

2.2.1 Realistic Input

Many systems modelling music cognition rely on manual preprocessing or annotation of performances before analysis by the system can take place. This information can include bar lines [27, 98], tonality annotations [27], and phrasing marks [27, 91]. However, as Jackendoff [61] states, the listener in reality only hears a sequence of pitches with durations, while the notated

key signature, time signature, bar lines and beams play no role in the musical surface.

Pre-annotation of the input with the above information, such as the standard music notation shown in Figure 2.2, ignores people's ability to extract information on their own from the performance. Therefore, a system attempting to more realistically model the listening process should take as input only pitch and timing information, without extra higher-level annotations. (As discussed in Section 1.4.2 above, other low-level audio qualities of music are not handled by Maestro.)

Some previous systems quantise the timing information into durations of eighth [98] or sixteenth notes [132]. Maestro takes as input only the musical surface: pitch information in semi-tones[1] and timing information that is not quantised into fixed note-durations. This presents the system with a more realistic listening task. Figure 2.3 shows an example of a Bach Chorale in the format used by Maestro. Maestro also performs all the segmentation on its own, based on perceptual cues present in the musical surface, as described in Chapter 3.

2.2.2 On-line Processing

When listening to music, people continually make analysis decisions, but are limited to utilising only the information they have heard until that point. Some computer music cognition systems rely on free access to the performance in order to accomplish their analysis tasks, (e.g., [31], and to a certain extent [132]). The entire performance can be scanned before the system makes any decisions. While this may lead to better speed or efficiency, in a more cognitively realistic system analyses should be performed on-line, with the data available only within a temporal window stretching back a certain distance from the present. Future data should not be available until it appears in the performance. Maestro is designed in accordance with this approach.

Jackendoff emphasises that a theory of musical perception should contain an account of how the listener applies certain principles in real time to derive abstract structures for a piece as it is being heard [61]. Maestro labels the performance on-line according to the structural interpretation suggested by its previous experiences.

[1] While there exist musical styles that utilise finer-grain pitch information than semitones, a majority of world music can be at least approximated by semitones, and so this simplification is not considered to be a major limitation.

2.2.3 Multiple-Step-Ahead Prediction

When listening to a piece of music, listeners use their previous musical experiences to guide their expectations about what is to come next [68]. The expectations can be of various time horizons, sometimes only predicting one note ahead of time, at other times many notes. The forecast horizon depends on the listener's past experiences and on the current musical context.

As a simple example, consider any multi-note repeating motif, such as the ones appearing in the well-known tune *Twinkle Twinkle Little Star*. After hearing the two seven-note motifs in the beginning of the piece, a listener knows what to expect at the end of the piece when the motifs begin to appear again. At that point multiple-step-ahead predictions may be generated. There are, on the other hand, instances when people can only predict one step ahead – for example, at the penultimate note of a repeating motif, or even zero steps ahead (no prediction) such as at the beginning of a new piece.

Previous studies [27, 132] on music prediction have focused on one-step-ahead forecasts. The present research studies the generation of appropriate multiple-step-ahead predictions in accordance with the current musical context.

2.2.4 Time and Memory Constraints

Humans have a limited capacity to process information in real time. For example, there is a three-to-five-second-long audio window memory containing information that disappears unless it is stored elsewhere [44, p. 180]. Also, people are limited in the amount of information they can remember after listening to a piece of music [115, p. 190]. Previous work in modelling music cognition has ignored some of these limitations [27, 72]. Maestro's design takes these constraints into consideration in such tasks as storing information into its model, and in performing on-line parsing.

2.2.5 Ambiguity Handling

While it is well known that ambiguity arises in the visual realm (Figure 2.4) ambiguity is also an essential part of the musical listening experience. In the paper *Functional Ambiguity in Musical Structures* [120], Thomson describes various properties of a musical event, including harmony, timbre, texture, form and melody, that can sometimes act together in building up implications in the listener. However, when these properties are not congruent with one another, conflicting implications are created causing ambiguity to arise. Dowling similarly states that ambiguity arises when there are conflicting tendencies among certain musical invariants [44, p. 192].

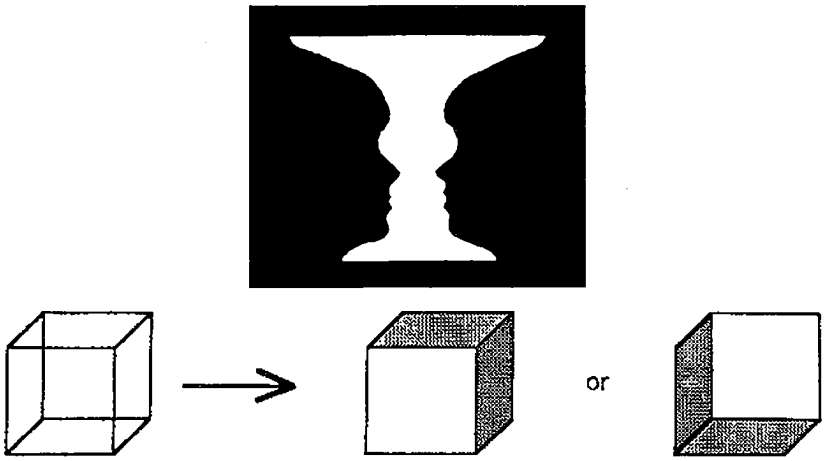


Figure 2.4: Examples of visual ambiguity. From (Thomson, [120]) and (Rosenthal, [98]).

Ambiguity often emerges out of the interrelated and simultaneous patterns present in the music. However, as Conklin and Witten [132] explain, it is sometimes placed by the composer intentionally to introduce elements of originality or surprise into a piece. Berent and Perfetti [8] state that ambiguous passages can also be used as gradual transitions between passages of different types. Kippen emphasises that there is a need for a better understanding and formalisation of ambiguity in musico-cognitive processing [66, p. 329]. The present research seeks to address this need by incorporating the handling of musical ambiguity into a machine model of music cognition.

Thomson provides examples of various types of musical ambiguity, two of which are shown in Figure 2.5. While there are many different types of ambiguity in music, this research focuses specifically on three types:

1. *Segmentation ambiguity* – A lack of clarity regarding how to break up the music for storage into memory. In Maestro, this results from conflicts in discontinuity-based segmentation, as discussed in Chapter 3.

2. *Prediction ambiguity* – A lack of clarity regarding what is to come next in the piece. Maestro's handling of prediction ambiguity is described in Chapter 5.

3. *Parsing ambiguity* – A lack of clarity about how to structurally interpret the piece once it has been heard. In Maestro this results from conflicts in repetition-based segmentation, as discussed in Chapter 6.

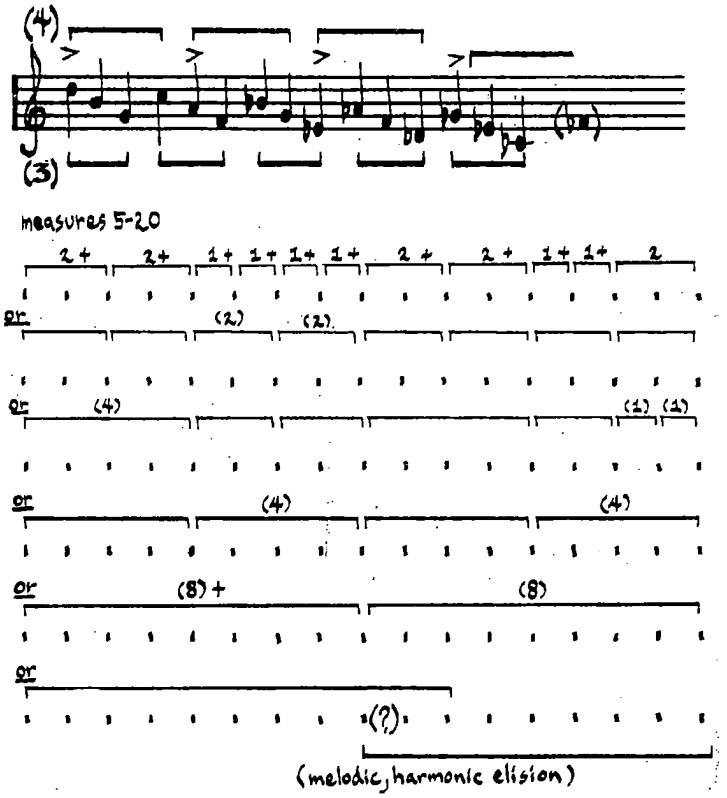


Figure 2.5: Examples of musical ambiguity. Top: Metric ambiguity caused by contradictory stress and contour patterns. Bottom: Six potential groupings for measures 5-20 of Chopin's *Mazurka*. From (Thomson, [120]).

Ambiguity is especially relevant in the context of music learning. A well-trained system has a large set of learned biases and intuitions that it can use to resolve ambiguity, while an untrained system has few. Davidson *et al.* report experiments showing that in an ambiguous grouping context, musicians tend to classify events according to higher-level schemata, while non-musicians tend to classify events based on more primitive grouping cues [37]. Similar results are cited by Sloboda [115, p. 187]. Maestro deals with modelling an untrained listener, and therefore relies on bottom-up perceptual cues present in the musical surface in order to resolve ambiguity.

Musical ambiguity often leads to multiple hypotheses of interpretation: more than one interpretation may be consistent with the music heard thus far. According to Jackendoff [61], in the absence of sufficient information during on-line listening, multiple hypotheses may have to be maintained until enough information becomes available to resolve the ambiguity.

However, almost all systems modelling music cognition ignore the ability of humans to maintain multiple hypotheses when facing ambiguity. Instead, either a judgement call is made by the human operator in the pre-processing stages, or the ambiguity is clarified by a fixed system of rules [98]. Either way, the ambiguity is immediately resolved, and only one interpretation is maintained by the system. The ability to generate, maintain and reconcile multiple hypotheses of interpretation is a central feature of Maestro.

Berent and Perfetti formulate what they refer to as the Parsing Problem: given the limitations of working memory, the listener must arrive at decisions almost immediately, but the input available to the listener at the time does not lend itself to a unique representation [8]. Berent and Perfetti note that this problem is, of course, solved by listeners. This *deterministic* approach to resolving ambiguity is dealt with in Maestro's parsing stage, as described in Chapter 6.

2.2.6 Retrospective Listening

Often at ambiguous points in a piece, a decision about the final interpretation is delayed. When further information becomes available, a final decision can be reached, and the listener goes back and "re-hears" the music according to the new analysis. According to Jackendoff, this results in the listener hearing the music with the proper structure, and projecting the structure backwards to the point where the ambiguity originally arose [61]. Jackendoff describes this phenomenon as "retrospective hearing" or "retrospective reanalysis," while Berent and Perfetti [8] call it "reinterpretation". A system capable of on-line handling of ambiguity should be able to exhibit this phenomenon. Maestro's parsing stage is designed to meet this requirement.

2.2.7 Active Listening

Listening is not a passive process. Rather, listeners actively engage with a musical performance, segmenting it, storing it in memory, generating expectations, building abstract structures from the musical surface, and comparing it with previous pieces they may have heard. Jackendoff points out that in order to make such a comparison, the processor has to be actively comparing the presented music with the recalled music [61].

In a similar vein, Reiss Jones *et al.* [65] state that remembering is assumed to involve a recapitulation of the original rhythmical activities that were involved in attending to a melody. Reiss Jones [64] puts forth a theory of *dynamic attending*, in which expectancy aids processing by directing attentional resources to the appropriate range of pitches at specific times. In discussing memory for rhythm, Rosenthal [98] states that memory and process are contained in one structure. As Minsky [82] explains, in order to understand how memory and process merge in listening, researchers have

to learn to use much more "procedural" descriptions. This procedural approach, in which musical memory is intrinsically tied with music processing, is also shared by both Blacking [18] and Hiraga [55], and is referred to here as the *active listening approach.* Maestro's design incorporates the multi-agent system paradigm to implement this concept, as described in Chapter 4.

The multi-agent system paradigm serves an additional purpose. When handling ambiguity, multiple active listeners must be present to simultaneously perform the above listening tasks according to the various interpretations being entertained at that time. Maestro accomplishes this by instantiating different active listening agents to represent different interpretations of the music. Each agent independently and actively monitors and responds to the music, generating predictions and attempting to parse the musical surface.

2.3 Summary

This chapter has reviewed the main features of Maestro's design. Maestro focuses on studying music learning, and in order to maintain the flexibility to handle music from many different styles, it includes no *a priori* style-specific knowledge. Additionally, large data sets are used in experiments to simulate a more realistic learning process.

Maestro's design is guided by principles from cognitive musicology. Analysis is performed on-line, and no extra information is present in the input. Maestro stays within certain realistic constraints of memory and timing, and generates appropriate multiple-step-ahead predictions according to the current musical context.

Ambiguity is an essential aspect of music listening, especially in the context of learning, and Maestro is designed to deal explicitly with three types of ambiguity, performing retrospective listening when appropriate. In order to implement the active listening approach and to deal with ambiguity, Maestro incorporates the multi-agent system into its design.

The next four chapters now turn to presenting a detailed account of the various stages of Maestro's design and operation, beginning with Maestro's segmentation stage.

Chapter 3

Segmentation

The purpose of segmentation, also called grouping or phrasing, is to divide the musical surface into salient chunks for interpretation and storage in memory. This chapter describes Maestro's segmentation capabilities. First, relevant background from cognitive musicology is presented and previous machine models are reviewed. Then, the implementation of Maestro's segmentation stage is described, including multiple segmentation modules and the handling of segmentation ambiguity. The concept of perceptually guided segmentation is presented as a basis for Maestro's approach. Finally, the circularity problem that is inherent to segmentation is introduced.

3.1 Background

> *Grouping can be viewed as the most basic component of musical understanding. (Lerdahl and Jackendoff, [72, p. 13])*

In the course of listening to a piece, a listener groups the performance into salient chunks, building up an internal representation of the piece's structure. Dowling discusses the *Gestalt* principles that come into play in music cognition, and includes proximity, similarity, common fate, and good continuation [44, p. 154]. The law of proximity dictates that elements grouped closely together on a particular dimension tend to be perceived as a single unit, separate from other more distant elements [115, p. 187].

People tend to find group boundaries at points of discontinuity; as Tanguiane states, all the known ways of accentuation are based on breaking the homogeneity in the music [118, p. 140]. Krumhansl and Jusczyk summarise the various cues that are used in musical segmentation, including contrasts of pitch range, dynamics, and timbre, lengthening of durations, changes of melodic contour, and metrical, tonal, and harmonic stress [69].

Lerdahl and Jackendoff put forth a detailed segmentation strategy in the grouping analysis portion of their Generative Theory of Tonal Music [72]. In [119], Thomassen outlines a model of melodic accent based on different types of discontinuity in pitch. In [59], Huron and Royal compare eight competing notions of melodic accent and state that their results are most consistent with the perceptual model of melodic accent developed by Thomassen.

Maestro segments according to discontinuities in time and pitch, and points of inflection in pitch. As Maestro does not handle polyphony, harmonic accents and points of dissonance are not addressed by Maestro's segmentation stage. Additionally, as Maestro does not maintain an explicit sense of meter, it does not segment according to strong metrical position, as suggested by Lerdahl and Jackendoff [72].

Hiraga [56] mentions that identifying repeating patterns, or parallelism, in music is also important for segmentation. This type of *repetition-based segmentation* is a necessary component of musical cognition [63], and leads to the circularity problem introduced at the end of this chapter. Repetition-based segmentation is handled by Maestro's parsing stage, as discussed in Chapter 6.

3.1.1 Previous Machine Models

Previous systems have addressed the question of music segmentation. A sample of relevant systems is now brought, presented in order of increasing levels of cognitive realism.

David Cope's *Experiments in Music Intelligence* (EMI) system, developed to compose pieces in a learned style, scans pieces of music in search of composers' characteristic patterns or *signatures*. EMI segments music in a fixed way, storing every chunk of n notes. The value n is set by the `pattern-size` controller [34]. Cope's approach ignores perceptual cues in segmenting the music. This is understandable, as Cope sets out to implement a computer system capable of generative style imitation, and not a cognitively realistic model of music listening.

Conklin and Witten's system [27], designed for their research in music prediction, takes in music that is pre-labelled with bar lines and fermatas, and these annotations are used to aid prediction. This input is not realistic for modelling music listening because, as mentioned in Section 2.2.1, these features do not appear in this ready form on the musical surface. Conklin and Witten's system then segments the music for storage, storing every possible chunk of up to length three[1] [27]. As with Cope's system, perceptual cues are ignored in segmentation. Conklin and Witten set out to develop a machine model of music prediction, not necessarily guided by cognitive principles.

[1] Two is the maximum length in Conklin and Witten's Short Term Memory model.

In his research on artificial perception, Tanguiane's system performs segmentation based on a series of timing accentuation rules formulated by Broda [118]. He discusses two levels of accentuation brought about by the presence of relatively longer notes. *Strong accentuation* occurs when a short note follows a long note, while *weak accentuation* occurs when two long notes appear together. Tanguiane's segmentation is thus based on temporal perceptual cues in the music.

Rosenthal's system for analysing rhythmic structures [98] takes in music that is pre-annotated with bar lines, with accents placed on the first notes in each measure. Rosenthal uses Lerdahl and Jackendoff's grouping preference rules [72], basing segmentation on time-based perceptual cues. Rosenthal's system deals with timing and rhythmic information and totally ignores pitch. However, Rosenthal notes the usefulness of pitch for segmentation, whereby large changes in pitch tend to accentuate certain notes, and states that an improvement to his system would utilise pitch information for segmentation. Rosenthal discusses segmentation ambiguity – situations where more than one segmentation is possible. In dealing with multiple segmentation solutions, Rosenthal's system chooses the solution whose length is closest to the previous chunk. While Rosenthal's system immediately resolves segmentation ambiguity, he suggests as a possible improvement that a system could keep multiple representations.

Of all the systems mentioned here, Rowe's Cypher system [102] is the one whose approach to segmentation is most based on perceptual cues. Segmentation in Cypher is performed by searching for salient discontinuities of density, register, speed, dynamic, duration, harmony, and beat. The individual discontinuities are calculated using the *focus and decay* method [102, p. 50], described below in Section 3.2.2. Once calculated, the presence of each of these various discontinuities contributes a certain weight to Cypher's total discontinuity measure in accordance with a fixed weighting scheme. When the accumulated discontinuity crosses a certain threshold, a segmentation boundary is declared [101, p. 61].

While Cypher is an impressive implementation of perceptually based segmentation, it has some limitations. First, Cypher incorporates an *a priori* knowledge of tonality and uses it to help in determining phrase boundaries. Second, when dealing with segmentation ambiguity, Cypher uses a fixed weighting scheme to combine all the perceptual cues and arrive at one segmentation, thus avoiding multiple segmentation hypotheses.

3.2 Segmentation in Maestro

Unlike previous systems described above, Maestro does not rely on pre-segmented input that is manually annotated with phrase markings, nor does

it incorporate any style-specific information. Instead, in keeping with the cognitive realism approach, it uses perceptual cues present in the musical surface itself to segment the performance into salient chunks. For this reason, Maestro's segmentation strategy is called *Perceptually Guided Segmentation* (PGS).

3.2.1 System Operation

The operation of Maestro's segmentation stage consists of three main steps:

- Input;
- Segmentation-point marking;
- Candidate Segment suggestion.

3.2.1.1 Input

Maestro receives its input in the form of a string of musical events, with pitch in semitones, and onset time and duration information (not quantised into standard note-lengths). See Appendix C for a detailed description of Maestro's data format. In keeping with the specifications set out in Chapter 2, Maestro processes this input on-line. In the data available for the experiments, information on dynamics (loudness) was either not available or was found not to be very useful for segmentation. Therefore, dynamics information is not dealt with in this research.

3.2.1.2 Segmentation-point Marking

Maestro identifies appropriate points of segmentation in the music by examining certain perceptual cues. Since there are multiple cues in the musical surface, three different segmentation strategies operate in parallel within Maestro's segmentation stage. These strategies are implemented as three separate segmentation modules (Figure 3.1):

1. Time Discontinuities – finds significant changes in inter-onset interval.
2. Pitch Discontinuities – finds significantly large leaps in pitch.
3. Pitch Inflection Points – finds significant changes in melodic direction.

The operation of each segmentation module is described in detail below in Section 3.2.3. When presented with input, a segmentation module marks points of segmentation on the musical surface. Segmentation marks are placed automatically at the beginning and end of a piece, ensuring that

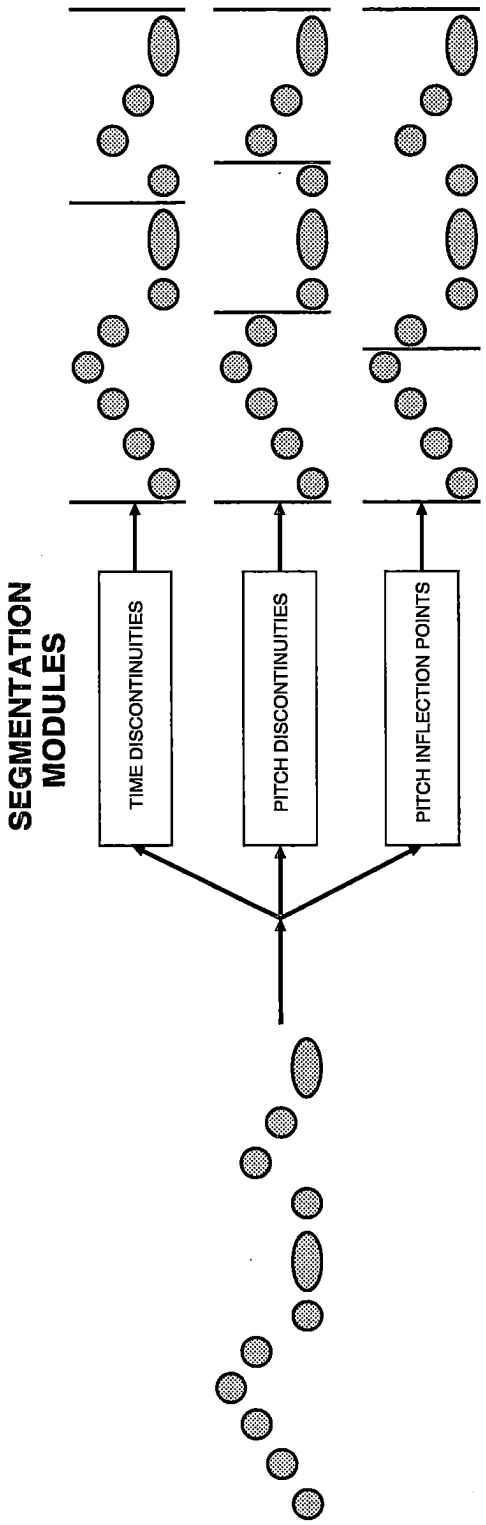


Figure 3.1: Maestro's segmentation stage, consisting of three segmentation modules.

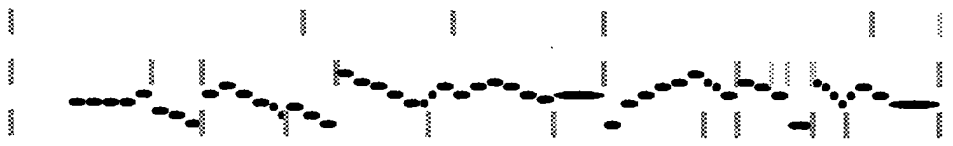


Figure 3.2: Maestro's segmentation of Bach Chorale Number 2. The three rows of bars represent, from top to bottom: Time Discontinuity, Pitch Discontinuity, and Pitch Inflection Points. Lighter bars indicate segments that are too long or two short, and thus are not suggested for inclusion in the model.

all events are presented to the model for inclusion. This is consistent with Jackendoff's approach of placing a group boundary at the beginning of a piece [61].

Recall that Rosenthal's system [98] stores only timing information, and therefore segments only according to timing information. In contrast, even though Maestro only stores pitch information, it makes use of pitch as well as timing information for the purposes of segmentation.

3.2.1.3 Candidate Segment Suggestion

A segmentation module generates a candidate segment from the notes lying between a specific segmentation point and the previous segmentation point. This portion of the melody is passed on to Maestro's modelling stage for possible inclusion in the context model.

In order to avoid very short or very long segments, upper and lower bounds are placed on segment length. A segment is only suggested if its length is greater than two notes and smaller than twenty notes. These bounds were chosen to cover a wide range of segment lengths in order to maintain flexibility across different styles.

Figure 3.2 shows Maestro's segmentation of a Bach Chorale using its three segmentation modules. Marks denoting segments of appropriate lengths are coloured darkly, while marks representing segments which are too long or too short, are drawn lightly. Even though only dark marks result in the segment being suggested for addition to the model, both dark and light marks are used as the beginning-points for the following segments.

3.2.2 Focus and Decay

Dowling notes that music is context dependent – local features have meaning based on global features [44, p. 171]. With regard to segmentation, Thomson [120] emphasises that phrase boundaries are determined by the events they help to phrase. Since Maestro's segmentation modules attempt to identify significant discontinuities at which to declare segmentation boundaries, they

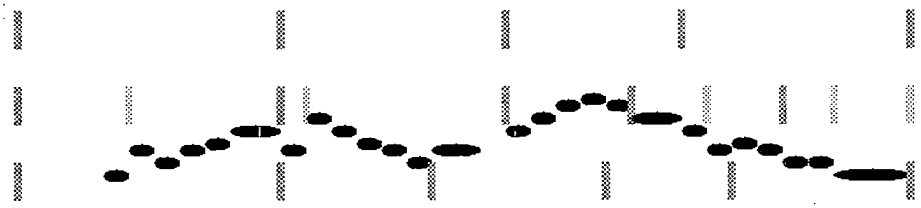


Figure 3.3: The segmentation of Bach Chorale Number 6, showing focus and decay operating in the pitch discontinuity segmentation module (middle trace).

rely on a definition of which intervals or changes are considered significant.

To address this issue in the first two segmentation modules, Maestro incorporates an adaptive approach based on the method of focus and decay, used by Rowe in his Cypher system [103]. The determination of relative significance in the third segmentation module is described below.

In the focus and decay methodology, the system keeps track of the maximum and minimum discontinuity magnitudes that it has seen so far. This range (between minimum and maximum) is then used to judge the relative magnitude of new discontinuities. With time, the minimum and maximum bounds slowly decay towards the centre of the range, but they can be pushed outwards again by a re-occurrence of an extreme value.

As a result of the adaptive nature of focus and decay, pieces with only very subtle discontinuities can still be segmented properly, while those with many large jumps will not trigger segmentation too often. This is in line with Maestro's focus on learning and flexibility, as Maestro must learn to notice what is significant in each specific musical context.

The implementation of the focus and decay concept in Maestro is slightly different from that of Rowe. Rowe maintains a small number of categories into which a certain feature, say pitch, is classified. These categories are stretched to cover the current maximum-to-minimum range. A discontinuity is then declared whenever the melody crosses from one category of pitch into another. Unlike Cypher, which maintains a range of pitches, Maestro maintains a range of pitch interval sizes. A discontinuity is declared when the current interval is greater than a certain percentage of the maximum interval seen so far, as described below. Maestro imposes additional conditions on declaring discontinuities, also discussed below.

Figure 3.3 shows focus and decay at work. The segmentation module tracking pitch discontinuity (middle trace) at first does not view a jump of two semitones as significant (e.g., after note 2 and after note 15). However, after enough exposure to the step-wise nature of the melody, the threshold

for pitch discontinuity significance decreases and the two-semi-tone interval after note 18 is labelled as a segmentation boundary.

3.2.3 Segmentation Modules

The specifics of each of Maestro's three segmentation modules are now discussed individually. Despite the design principle stressing flexibility, certain values had to be chosen as initial settings. In an attempt to maintain flexibility, the specific numbers reported below were set by trials with music from different styles. Furthermore, many of the parameters are adaptive, so the settings affect only Maestro's handling of the beginnings of each piece.

3.2.3.1 Time Discontinuities

The first segmentation module examines the inter-onset interval (IOI) – the difference in time between the starting points of two adjacent notes. Note-duration is not examined directly, as it is included within IOI. Tanguiane also looks at IOI and not at note-duration [118].

A set of maximum and minimum IOI values is maintained, and set initially to 1.0 and 2.0 respectively. The maximum and minimum values are decayed by 10 percent for every event processed. This is a much faster decay rate than is used by Rowe, who decays the boundary values by one semi-tone every five seconds [102, p. 52].

Two conditions must be met for a discontinuity to be declared by this segmentation module:

1. The current IOI must be greater than 80 percent of the maximum IOI.
2. The current IOI must be greater or equal to twice the previous IOI.

From test trials with segmenting music from various styles, both conditions were found to be necessary. If only the first condition is present, a series of long notes in succession repeatedly triggers the segmentation module. Alternatively, if only the second condition is present, a spurious discontinuity is declared whenever a relatively short note is followed by a note of intermediate length. Only by combining the two conditions are these situations avoided. The first condition is an implementation of focus and decay. The second condition is in line with Hiraga's statement that a logarithmic basis for comparing note length is prevalent, as in common music notation [55].

3.2.3.2 Pitch Discontinuities

The second segmentation module is similar to the first, but deals with pitch instead of time. It examines the absolute pitch interval – the absolute dif-

ference between the pitches of two adjacent notes. A set of maximum and minimum absolute pitch interval values is maintained, and set initially to 0 and 5 semitones respectively. As with the first segmentation module, the maximum and minimum values are decayed by 10 percent for every event processed.

Two conditions must be met for a discontinuity to be declared:

1. The current interval must be greater than 80 percent of the maximum interval.
2. The current interval must be greater than the previous interval.

From trials with segmenting music from a variety of styles, both conditions were found to be necessary. If only the first condition is present, a series of large intervals in succession repeatedly triggers the segmentation module. Alternatively, if only the second condition is present, a spurious discontinuity is declared whenever a relatively small interval is followed by a medium interval. Only by combining the two conditions are these situations avoided.

3.2.3.3 Pitch Inflection

The third segmentation module looks for changes in melodic direction, otherwise known as points of inflection. An internal counter keeps track of how many steps the melody has taken in the same direction (up or down) since the most recent change in direction. Repeated notes (unisons) are ignored by the counter.

The counter is initially set to zero. If three or more steps have been taken in the same direction, a change in melodic direction causes a discontinuity to be declared and the counter is reset to zero. If fewer than three steps have been taken before a change of direction, the counter is simply reset to zero and no discontinuity is declared.

Based on trials with music from different styles, the threshold of three steps was found to prevent spurious inflection points from being declared where the melody changes directions while no clear melodic direction has been established.

3.2.4 Multiple Segmentations and Ambiguity

The usefulness of the three cues used by Maestro varies between different pieces and different styles: for specific pieces, some methods of segmentation work better than others. For example, in Figure 3.3 the time-based segmentation module (top-most trace) seems to yield the most intuitive segmentation of the piece, breaking it up into similar-sized portions. However,

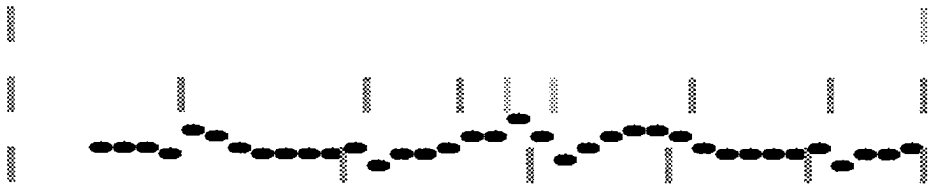


Figure 3.4: Maestro's segmentation of Bach Chorale number 13.

this time-based segmentation proves relatively useless in Figure 3.4, as the melody there contains no temporal accentual information.

Maestro strives to maintain the flexibility to handle music from many different styles, thus relying on a bottom-up approach. By analysing all these perceptual cues, (as opposed to, say, only pitch like Hiraga [55]), Maestro maximises the use of the information available in the musical surface. However, these multiple sources of information often suggest conflicting segmentation hypotheses [120]. Meyer suggests that segmentation ambiguity can be composed intentionally to serve a double purpose – Sloboda explains that ambiguity can lead to the double result of articulation together with forward-movingness by putting the various segmentation cues out of phase with one another [115, p. 190].

Maestro handles segmentation ambiguity by storing all the segmentation possibilities, which are then used to construct a model of the data. This approach is recommended by Rosenthal [98] (citing [74]) as a possible improvement to his system, which only stores a single segmentation interpretation. This is also in line with Jackendoff's general approach of handling ambiguity by maintaining multiple hypotheses [61]. While Rowe [103] combines all the possible segmentation hypotheses into one interpretation, Maestro maintains them as separate hypotheses, storing multiple segmentation interpretations into its model.

3.2.5 Perceptually Guided Segmentation

In anticipation of the discussion of Maestro's modelling stage in the next chapter, the issue of music segmentation is now addressed from a different perspective: modelling efficiency and cognitively realistic modelling.

Consider the pattern *ABCDEF*, seen in the data. A model can store *ABC, BCD, CDE* and *DEF*, capturing all possible three-note segmentations of the data. Alternatively, a model can disallow overlapping segments and instead store only *ABC* and *DEF*. Conklin and Witten [27] take the former approach, *full-overlap*, while Rowe [102] and Cope [34] adopt the latter, *no-overlap*. Figure 3.5 shows examples of these different approaches.

Each approach has its own advantages. The more contexts that are

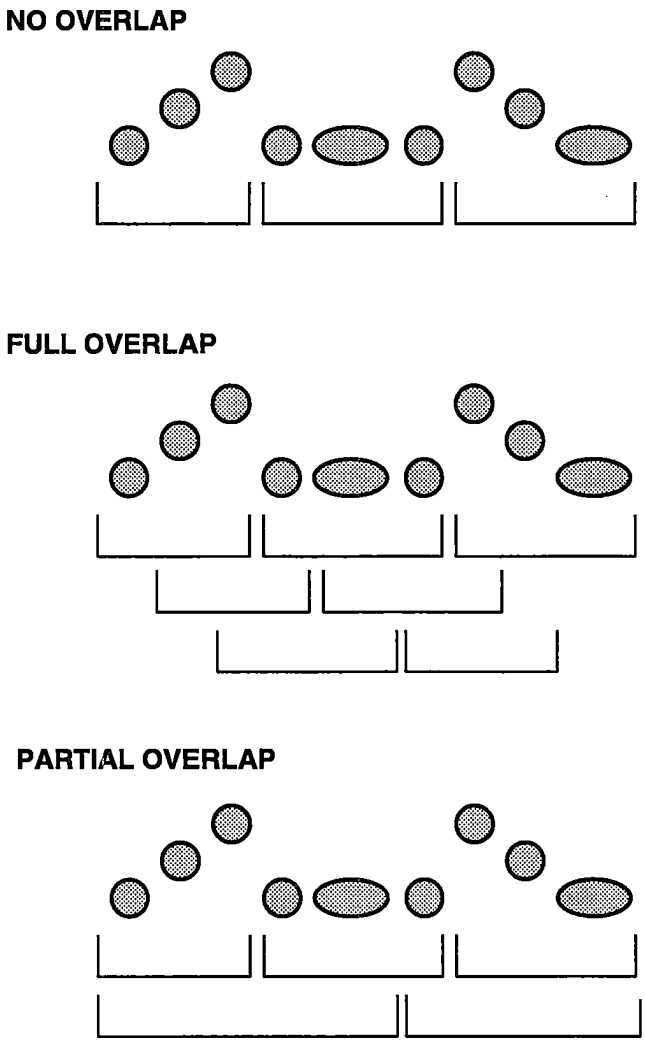


Figure 3.5: Examples of three alternative approaches to segmentation and modelling. From top to bottom: no-overlap, full-overlap and partial overlap.

stored, the greater the robustness of the model for the purposes of prediction. However, a full-overlap strategy can also lead to unacceptably large model sizes, pushing the limits of memory. There is thus a trade-off between model robustness and model compactness.

It is highly unlikely that people redundantly store all possible segments up to a certain length when listening to a piece of music. Conklin and Witten set out to construct an optimised machine model of music prediction, and cognitive constraints are not a primary concern in designing their full-overlap model. However, the present research is aimed at developing a more cognitively realistic model of music segmentation.

An intermediate approach is proposed here which offers a beneficial compromise between model robustness and model compactness, and also leads to more cognitively realistic segmentation and modelling. As mentioned above, Perceptually Guided Segmentation (PGS) is the name given to the segmentation implemented in Maestro, based on multiple perceptual cues in the musical surface. Since the various perceptual cues do not always coincide with each other, multiple possible segmentations are often suggested. In such cases of segmentation ambiguity, as suggested by [98] and [74], all segmentation possibilities are stored, leading to *partial overlap* in the model (Figure 3.5, bottom).

This approach is more space-efficient than a full-overlap model, which ignores perceptual cues and stores every possible segment. In accordance with the aforementioned trade-off between size and robustness, with this improvement in size-efficiency, one can expect a decrease in robustness and prediction performance compared to the full-overlap strategy.

However, it is proposed that this expected drop-off in the robustness is somewhat mitigated by the efficiency inherent to the perceptually guided nature of the approach. The claim is that the segments delineated by perceptual cues and selected using the PGS strategy are correlated with the structurally salient patterns in the music. It is further proposed that these structurally salient patterns are the ones most likely to repeat later in a piece, and are therefore likely to be more useful for prediction. Therefore, by storing only the more relevant segments, model efficiency is increased, and a greater degree of cognitive realism is maintained.

This concept that some segments are more relevant than others is mentioned with varying degrees of explicitness in the literature. When discussing segmentation, Sloboda states that listeners use certain cues to segment music into manageable and "meaningful" short units [115, p. 190]. Rowe also hints at this idea implicitly in a 1994 conference paper where he states that input is segmented into phrases in the hope that pattern instances will begin at phrase boundaries [101, p. 60]. Finally, Cambouropoulos explicitly suggests that "perceptually-pertinent" local discontinuities should be used

in guiding the search for "significant" patterns in the music [20, p. 46]. While this concept has been mentioned in the literature, to the author's best knowledge it has not been empirically tested.

Apart from the widely-supported view that people use perceptual cues to segment music, there is also some evidence that people actually *store* the music in this way. Sloboda reports experiments that studied subjects' ability to recall patterns found in different parts of a tune. Some patterns were taken from within the perceptually-delineated phrases in the tune, while other patterns "straddled" phrase boundaries. The results indicate that listeners were more likely to form accessible memory representations of intervals *within* phrases. [115, p. 190]. Maestro's strategy of constructing the model directly from the results of the perceptually guided segmentation process is in line with these findings.

Finally, in addition to resulting in more efficient, cognitively realistic models for prediction, perceptually guided segmentation also fits better with the "learning by listening" approach mentioned in Section 1.2.3, as learning (storing segments in the model) happens as a direct result of listening (segmenting the musical surface).

It is thus hypothesised that PGS allows Maestro to capture the intrinsic structure of the music, and therefore leads to more efficient models for prediction in light of the size constraints imposed by cognitive realism. To validate these hypotheses, an experimental method called *N-Note Segmentation Shifting* is developed and experiments are performed, as described in Chapter 9.

3.2.6 The Circularity Problem

Maestro segments the musical surface according to local discontinuities and points of inflection. Hiraga [56] points out that while this approach is effective in a wide variety of musical settings, it cannot handle certain instances in which segmentation must rely on identifying parallelism (repetition of patterns) in the music.

Repetition-based segmentation uses the re-occurrence of previously seen patterns to inform the segmentation process. However, knowing which patterns to search for usually relies on some sort of segmentation to generate a set of patterns from the input stream, as a full search for all possible patterns is not usually feasible. Therefore, segmentation relies on spotting repetitions, while spotting repetitions in turn relies on segmentation. The inherent circularity of this problem is noted by Hiraga [56] and Larson [70].

Maestro deals with repetition-based segmentation in its parsing stage, and the circularity problem is addressed in full detail in the discussion of parsing in Section 6.2.4.

3.2.7 Comparison with Cambouropolous

The work of Emilios Cambouropoulos, specifically the Local Boundary Detection Model (LBDM) [19] is relevant to this discussion. Cambouropoulos describes a system for segmenting musical input based on the Gestalt principles of proximity and similarity, identifying discontinuities in both pitch and time signals. While there are significant similarities with the present work, the primary difference lies in the approach. Cambouropoulos' LBDM determines boundaries based on a local context of a few notes at a time. In contrast, Maestro adopts the Focus and Decay approach which considers a much larger context in making boundary decisions. Cambouropoulos' work is considered again in Section 6.2.4 during the discussion of the circularity problem.

3.3 Summary

Segmentation is a crucial component of music cognition. Listeners must break up the musical input to interpret a performance and store it in memory. Evidence from cognitive musicology indicates that segmentation relies heavily on perceptual cues in the musical surface. A number of previous machine models, such as Cope's and Conklin and Witten's, ignore perceptual cues in performing segmentation.

In keeping with the general focus on cognitive realism, Maestro's segmentation stage operates based on bottom-up perceptual cues in the musical surface. Three perceptual cues are monitored by the individual segmentation modules: timing discontinuity, pitch discontinuity, and points of inflection in pitch. In determining the relative significance of various discontinuities, a modified form of the focus and decay strategy used by Rowe is incorporated to allow Maestro to adapt to the changing musical context.

Various cues can often lead to conflicting segmentation possibilities. In contrast to previous systems, Maestro stores all possible segmentations in its model. This is in keeping with Jackendoff's general approach of maintaining multiple hypotheses when faced with ambiguity.

It is proposed that Maestro's perceptually guided segmentation strategy, apart from being more cognitively realistic, leads to more efficient models for prediction than other segmentation strategies.

Repetition-based segmentation and the circularity problem inherent to it are dealt with fully in Chapter 6, which deals with Maestro's parsing stage.

Chapter 4

Modelling

The purpose of modelling is to construct and maintain an internal representation of what the system learns from its musical experiences for future use. This chapter describes Maestro's modelling capabilities. First, the relevant issues in cognitive musicology are presented, including a discussion of musical memories and musical storage formats. Then, different approaches to storing music information are considered and context models are formally introduced. Finally, Maestro's modelling stage is described, including the concepts of more cognitively realistic context models and of agent-based activated modelling.

4.1 Background

> *The way one hears music is crucially dependent upon what one can remember of past events in the music... A note or chord has no musical significance other than in relation to preceding or following events. (Sloboda, [115, p. 174-5])*

When listening to a piece of music, people store information for use in understanding that piece as well as in understanding future pieces. This information is stored in different types of musical memory.

4.1.1 Musical Memory

As information accumulates in memory over time, the listener's understanding of a piece or of a style gradually increases. Therefore, the storage of information in musical memory forms the basis for musical learning. It is widely held that there exist at least three types of musical memory: Echoic Memory (EM), Short Term Memory (STM), and Long Term Memory (LTM). As

Figure 4.1: The three types of musical memory modelled in Maestro.

their names suggest, these types of memory have different time-spans and can be used for different purposes. Each of these is now described in turn.

4.1.1.1 Echoic Memory

Echoic memory keeps track of items heard in the very recent past, and accounts for people's ability to internally review an audio snippet shortly after hearing it. Berz [9] describes a "musical inner speech" which is stored in a "Music Memory Loop", and Fraisse and Turner and Poppel (in [98]) describe a small auditory buffer of around 3 seconds. Dowling states that the audio memory usually contains information from the most recent 5 seconds, but can extend as long as 10 seconds [44, p. 180]. This information represents the psychological present. If people are asked to remember longer things, they must first break them up into shorter segments. For this reason, Dowling states that phrase lengths in songs are commonly about 2.5 - 5 seconds long.

Carroll-Phelan and Hampson [22] describe an "auditory buffer" for the brief retention of relatively unprocessed auditory input . This accounts for people's ability to regroup recent musical events into phrases after already having heard them, and thus forms the basis for the retrospective listening phenomenon described in Chapter 2. Maestro's parsing stage incorporates this concept of echoic memory in its ability to perform parsing retrospectively, as described in Chapter 6.

4.1.1.2 Short Term Memory

Short term memory is dynamic in the sense that it adapts to a particular sequence [27]. STM keeps track of items that are particular to the current piece being heard, such as themes and repeating patterns. This information falls into Narmour's category of intra-opus style. Berz cites evidence that STM for music is at least 180 seconds in length [9]. Both Rowe [103] and Conklin and Witten [132] assign STM the time-span of the current piece, and Maestro adopts this approach.

4.1.1.3 Long Term Memory

Long term memory can store information between listenings, and is associated with Narmour's extra-opus style. Stylistic rules or trends that govern items such as cadences or common progressions fall within the scope of LTM. High level stylistic information seems to be associated with long term memory, and Berz cites evidence that people are able to perform certain high level processing with musical information stored in LTM, but not with that stored in STM [9].

When listening to a piece from a particular style, information about the style is stored and updated in LTM. Therefore, with each listening, the ability to understand and attend to more of the complexities of the style increases. In this way, LTM forms the basis for the learning of musical styles. Both Rowe [103] and Conklin and Witten [132] maintain separate models for STM and LTM, as does Maestro.

4.1.1.4 Episodic, Semantic and Working Memories

Dowling draws a further distinction: *Semantic memory* records information about global invariants that are generally true for a style, while *episodic memory* stores specific local features and accounts for people's ability to remember specific selections of music, such that the next time that particular piece is heard, it is recognised as familiar [44, p. 165].

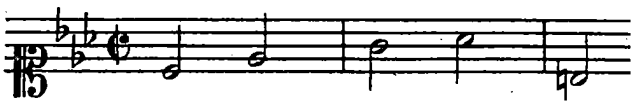
The term *Working Memory* is sometimes used synonymously with echoic memory, and sometimes with short term memory. However, it usually refers to the set of resources utilised for performing certain tasks of memory integration and look-up. It can be thought of as a run-time sketchpad for performing calculations that rely on the current musical input as well as information stored in other memories. The interested reader is referred to [9] for a recent theoretical model of working memory.

4.1.2 Forgetting

Almost as important as learning in music, is forgetting. Over time, the presence of certain items fades in memory. Forgetting affects the various types of memory differently.

In echoic memory, information is stored for only a few seconds. Whatever is not stored elsewhere (i.e. in STM or LTM) within that time is lost. This is implemented in Maestro's parsing stage, where ambiguity can cause the parsing of the music to be delayed for a short while, but not indefinitely, as described in Chapter 6.

With short term memory, information is stored only for the duration of a listening. At the end of a piece, or within a few minutes, the information is forgotten, unless it is stored in long term memory. According to Berz, information held in STM is easily lost if not rehearsed [9]. Conklin and



1. `Up, Up, Up, Down.`
2. `+3, +4, +1, -10.`
3. `E flat major; Begin on 6th scale degree; +2 +2 +1, -6.`

Figure 4.2: Three alternative melodic storage formats of the opening of J. S. Bach's *Muiskalisches Opfer (BWV 1079).* 1. Melodic contour, 2. Intervallic representation (semitones), 3. Scale-step representation. Score taken from (Fuller, [48]).

Witten explain that the short term model is transitory in the sense that it is discarded after each song [27, p. 57]. Maestro's implementation follows this approach.

In LTM, information is lost over time. Rowe's [102] system is designed so that the strength of patterns stored in long term memory fades if they are not reinforced through repetition. For reasons explained below in Section 4.4.1.2, Maestro does not implement forgetting in long term memory.

4.1.3 Format of Storage

There are different theories concerning how musical information is stored in the various types of memory. Sloboda [115] considers three possible formats for storing melodies, examples of which are shown in Figure 4.2:

1. Melodic contour – directional information of melodic progression.
2. Intervallic representation – exact information about both the direction and magnitude of the pitch intervals.
3. Scale-step representation – information about the relative intervals in terms of scale degrees.

It is well known that people can recognise melodies starting at different pitches (transpositions) and played at different tempos [44, p. 128], and so a format of musical storage must allow for this. In considering the above three possibilities for musical storage formats, Sloboda discusses experiments in which subjects are asked to compare different types of musical transpositions. Sloboda states that when people store melodies, they record the initial key and the relative scale degrees of the following notes. Dowling

[44] similarly suggests that people use an internal sense of musical scale to store melodies in a movable *do re mi* format. He says that maintaining an internal sense of scale allows the listener and composer to have the same schematic model, which aids in communicating the message of the music. As Maestro does not maintain an explicit sense of tonality, it cannot store scale-based information.

While Sloboda points out that melodic contour is "too crude" [115, p. 183] for most purposes, both he and Dowling note that storing melodic contour can still be useful in certain contexts – particularly in atonal settings where a clear tonal scale system is not present, or in the early parts of tonal pieces when the key has not yet been established [44, p. 134]. Dowling concludes that people use a combination of both scale steps and contour in remembering a melody. However, when it comes to storing information longer term, both Sloboda [115] and Dowling [43, p. 140] point out that exact intervals are used in place of contour, as people have the ability to distinguish between many similar-contour melodies they have heard in the past.

In evaluating the various storage possibilities, it is also important to consider developmental factors. People attain increasingly complex musical capabilities at different stages over the course of development. It seems that young children, who do not yet have a developed sense of tonality, rely more on contour information. As they grow, their ability to handle tonal information gradually develops, and they come to rely more and more on tonality in processing and storing musical information. Dowling reports that by age five, children have a sense of the tonal centre in a piece [44].

In [42], Dowling reports empirical evidence that inexperienced listeners represent melodies as sequences of pitch intervals that remain invariant across context shifts, while more experienced listeners appear to represent melodies as scale-step sequences that are affected by context. Since Maestro is intended as a model of an inexperienced listener undergoing music learning, it stores melodies in the form of pitch intervals, which allows for recognising melodies despite pitch transpositions or tempo changes.

4.2 Machine Modelling

A number of issues arise in designing a machine model of musical information storage that is guided by cognitive principles. Maestro's model is used primarily for parsing and prediction, and these two purposes are kept in mind when addressing the issues of modelling approach, representation format, and model type.

4.2.1 Approach to Modelling

The first issue in designing a machine model is the modelling approach. There are two common approaches to prediction: statistical time-series methods, and model-based methods [76, 108]. Time series methods look for trends in the data and attempt to extrapolate them into the future. In contrast, model-based approaches approach the task by developing a model of the source that generates the data and then using this model to predict what will come next. Since music cognition is widely believed to rely on some forms of internal models of musical structures, the model based approach is chosen for this task.

4.2.2 Representation

The second issue that needs to be addressed when designing a model is how the data will be represented in the model. In a survey of musical representation systems [130], Wiggins *et al.* state that proper evaluation of musical representation systems should focus on the particular purpose for which a representation will be used.

With this in mind, a major requirement of the present research is the flexibility necessary for learning to listen to music from different styles. A customised internal representation [95, 127] typically embodies certain assumptions and thus restricts the flexibility of a model. In order to maintain the generality of the model, Maestro instead uses the performance as a model of itself, storing segments of the musical surface. This is in line with Brooks' theory's of *intelligence without representation* [15], in which systems handling information from the environment avoid symbolic processing and instead use the environment as a model of itself. This general approach, also adopted by Bryson *et al.* [17], leads to system robustness in real-world noisy environments where hard-coded representations often prove too rigid and fragile.

The only processing that Maestro performs on the music is the conversion of absolute pitches to pitch intervals. As mentioned above, this allows for recognising melodic fragments despite transpositions, a skill known to be present in human listeners. Cope [34], Rowe [104], and Conklin and Witten [132] also follow this approach. Other options for machine representations of melody are reviewed by Cambouropoulos in [19].

4.2.3 Model Type

The third issue to consider is then: What type of model should be built? The answer clearly depends on the type of environment that needs to be

modelled – in this case, a melody. Maestro's goal of enabling a study of music learning requires that a model have the flexibility to learn music from different styles, while maintaining the clarity of system operation necessary for a careful study of the learning process.

Symbolic rule systems are generally rigid, and their flexibility in handling music from different styles is limited by the initial design (see examples in [72, 127] and a review in [97, Ch. 19]). On the other hand, connectionist systems offer more flexibility, but less clarity of system operation (see examples in [10, 51] and a very recent and comprehensive review in [52]).

In contrast, the flexible yet clear structure of Markov models (explained below) serves the purposes of this research well. Various types of Markov models have been used in different ways for music composition, as reviewed by Ames [4].

One final issue relating to modelling is now addressed: How complex should a model be? A few options are considered. A *Markov-1 environment* is one in which the next state relies solely on the information contained in the previous state [96]. Such a model allows for predicting musical events based only on the previous event received. This may be a good start, but is probably too limiting, as people can clearly use more information in making musical predictions.

A *Markov-k environment* is one in which the present state can be determined by examining the previous k steps. This is a more reasonable description of melodies, as it utilises the information present in larger musical structures for prediction. To model a Markov-k environment, a *k-order context model* can be used.

More complex models can be considered. Finite state machines, stack-based models or even general Turing machines are capable of performing complex operations on stored musical data [96]. It is not clear that all these capabilities are present in people's musical abilities, let alone that they are useful for the musical purposes of the present research.

Therefore, the modelling approach used here is the construction of context models from segments extracted from the music. Context models, otherwise known as $\mathcal{N}$*-grams*, have been used for text prediction [36], text compression [7], and game-playing (Pierre-Yves Rolland, personal communication). Context models have also been used for musical purposes. Conklin and Witten have conducted extensive research on musical prediction with context models of Chorale melodies (most recent in [27] and [132]), and have shown their machine results to be highly correlated with human data. Brooks *et al.* [14] also designed a system to learn Chorale melodies with context models, this time for the purpose of composition. Rowe uses a form of context model for automatic accompaniment in his Cypher system [103]. In more recent work, Ponsford *el al.* [91] use a context model to

learn abstract stylistic harmonic progressions for the purpose of music composition in the learned style. For the present research, the context model paradigm is chosen as appropriate for the task at hand.

4.3 Context Models

Context models are a subclass of the probabilistic finite-state, or Markov class of grammars [27, p. 55]. They store previously seen sequences and use them to predict future data values. A context model consists of three parts:

1. A database of sequences;
2. A frequency count attached to each sequence;
3. An inference procedure used to make predictions from the database.

Context models can learn with experience. When a new pattern of pitch intervals is noticed in the input, it is added to the context model. A frequency count is maintained for each pattern in the model, and the repetition in the input of a pattern already stored in the model causes its frequency count to be incremented.

In Maestro, each stored sequence is used as a context for the purposes of matching incoming musical data and generating appropriate expectations. For example, after the sequence ABC is seen and stored in the database, the appearance of AB in the input will lead the model to predict C as the next event.

Frequency counts are used as weights to integrate the various predictions generated when multiple contexts are activated simultaneously. In this way the context model can generate probabilities for different events occurring. Learning with context models involves either adding new segments or incrementing frequency counts of existing segments. These are referred to as structural and statistical learning respectively [27, p. 56]. For this reason, according to Conklin and Witten, induction of context models can be viewed as a hill-climbing search of a specialisation hierarchy of probabilistic theories.

As mentioned in Section 2.1.2, the present research rejects the knowledge engineering approach due to its dependence on the biases of the designer. In contrast, context models are totally inductive, and the resulting model is solely the product of the system's musical experiences. According to Bell *et al.* [7, p. 18] adaptive models discover for themselves the regularities that are present, and do so more reliably than people. They are not influenced by prejudice or preconception, but instead, their preconceptions are dictated precisely and predictably by the kind of data used to train them.

The length of the segments stored in the model is referred to as the model's *order*. Most context models have fixed-order – all stored segments are of the same length. In determining which order is best suited for a particular task, two important trade-offs arise: model size and model generality.

4.3.1 Model Size Trade-off

The first trade-off in choosing the order of a context model concerns the model size, and affects the training speed and prediction performance of the system: Shorter contexts make use of less information to generate predictions and thus generally lead to lower quality prediction performance. Still, these lower-order context models remain small and are easy to train.

Conversely, longer contexts lead to better prediction, since the model is said to have better *resolution* [7]. For example, using the context $ABCDE$ to predict E after seeing $ABCD$ will be more reliable than using the context DE to predict E after seeing D. However, due to to the exponential increase of model size with increasing order, the model becomes much larger and takes much longer to train. This trade-off is referred to as the *best-match problem* in [7, Ch. 3].

4.3.2 Model Generality Trade-off

Apart from the issues of training time and model size, it is also important to consider the generality of the model after learning. Conklin and Witten [27] explain that very low order models are too general, and do not capture enough structure of the concept, while very high order models are too specialised to the examples from which they were constructed, and do not capture enough statistics of the concept.

4.4 Modelling in Maestro

Maestro's modelling is a direct result of its segmentation strategy: after segmenting the music according to perceptual cues, Maestro stores the delineated musical information in its model. Dowling states that memory for a piece is organised into a set of episodes of varying length, and that phrasing is sometimes complicated by the overlap of perceptual cues [44, p. 165]. This is similar to the way in which Maestro's segmentation leads to modelling. In keeping with the general goal of using cognitive principles to guide Maestro's design, Maestro's modelling stage is an attempt to develop a more cognitively realistic context model.

4.4.1 Context Modelling: Implementation

Maestro maintains separate context models for short term memory and long term memory. Echoic memory is addressed during the parsing stage, as described in Section 6.2.2.3. Model-building in Maestro consists of two main steps:

- STM construction;
- STM to LTM roll-over.

4.4.1.1 STM Construction

Maestro's segmentation modules suggest various segments to the modelling stage for possible inclusion in the STM context model. If a suggested segment does not exist in the STM context model, it is added as a new segment. If an identical segment already exists, the frequency count of the existing segment is incremented. There is no minimum threshold for the number of repetitions necessary for inclusion in the context model, as suggested by [7].

4.4.1.2 STM to LTM Roll-over

At the end of each piece, Maestro transfers the contents of the STM model to the LTM model. This process again involves checking for duplicates to determine whether new segments should be added to the model, or whether the frequency counts of existing segments should be incremented.

The STM context model is completely reset at the start of each new piece, and thus forgetting takes place in short term memory. This is similar to the approaches of both Conklin and Witten [27] and Rowe [103]. However, like Conklin and Witten, but unlike Rowe, Maestro does not implement forgetting in long term memory. This would involve choosing a decay rate at which old, unreinforced patterns fade from memory. As the present research attempts to conduct a systematic study of music learning, an arbitrarily chosen decay rate would have a direct influence on music learning, affecting the results in an arbitrary way. Therefore, Maestro ignores forgetting on the long term level. This is especially relevant to the multi-style experiments described in Chapter 10, where experience with a previous style very much affects listening to a new style.

4.4.2 More Cognitively Realistic Context Modelling

Sloboda notes that the strategy of storing all melodic segments can quickly run into memory limitations. However, by capturing repetitions in the music, storage can be made more efficient [115, p. 190]. The context modelling

methodology is in line with this approach, and patterns that repeat are stored only once.

As mentioned before, Conklin and Witten report some impressive results with using context models for predicting Bach chorales [132]. However, Conklin and Witten's is an optimised machine-based approach, as opposed to a cognitively realistic one. Therefore, their system has a number of limitations with regard to the specifications of cognitive realism pursued in the present research. Conklin and Witten's system ignores perceptual cues in segmenting the music, and instead uses an exhaustive method that stores an unrealistic amount of information after each listening. Moreover, Conklin and Witten encode a set of style-specific knowledge into their system to assist in prediction, ignoring issues of pan-stylistic potential and music learning.

The present research seeks to extend the context modelling-based approach of Conklin and Witten and make it more cognitively realistic. As mentioned in Chapter 3, this is accomplished by allowing the lengths of the segments stored in the context model to vary according to the perceptual cues mentioned in Chapter 3. In this way modelling is intimately connected with segmentation. This approach maintains the model's flexibility to vary the segment length appropriately, according to the current musical context. This also leads to Maestro's ability to generate appropriate multiple-step ahead predictions, as discussed in the next chapter.

Maestro's modelling strategy addresses the model size trade-off. As mentioned in the previous chapter, a key assumption in Maestro's design is that due to the relevance of the initial segmentation of the data, all the possibilities for segmentation do not need to be considered for the purposes of prediction. This keeps the model size down by storing only the more relevant segments of various lengths, without having to fill in the entire exponentially expanding model space. Experiments performed to validate this approach are reported in Chapter 9.

The model generality trade-off is also addressed in that contexts of various lengths are stored, thereby not biasing the model towards being too specific, nor too general.

Although not called as such explicitly, Rowe's system also incorporates a variable order context model based on perceptual cues to aid in automatic musical accompaniment. However, as discussed in Chapter 3, Rowe does not store multiple segmentation hypotheses, and his system incorporates a style-specific sense of tonality.

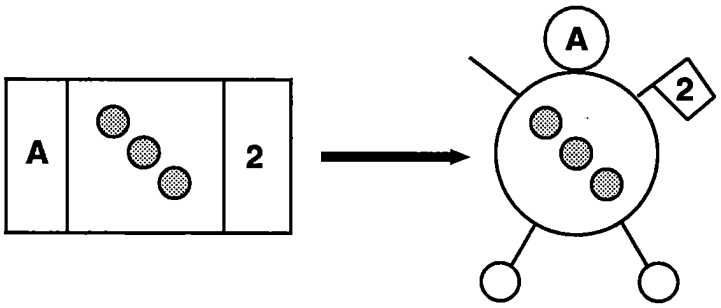


Figure 4.3: Activated modelling. A context model segment labelled A is shown (left), containing a pattern of three notes that has occurred twice (frequency = 2). When this segment is activated, a listening agent is instantiated (right), having the same label, the same three note pattern and the same frequency.

4.4.3 Activated Modelling: Listening Agents

Maestro is based on the principle of active-listening discussed in Section 2.2.7. Maestro also follows Jackendoff's [61] approach of maintaining multiple hypotheses when dealing with ambiguity. In implementing these two approaches, Maestro incorporates the multi-agent system paradigm into its design. Agents and multi agents systems were briefly introduced in Section 1.3.3.

Unlike the works of Rowe [103], Conklin and Witten [132], Brooks *et al.* [14] and Ponsford *et al.* [91], the context model stored in Maestro's modelling stage is not used directly for the purposes of prediction and parsing. Instead, the model is *activated*, as illustrated in Figure 4.3.

The activated modelling approach involves four steps:

- Segment activation;
- Agent instantiation;
- Agent-based prediction;
- Agent-based parsing.

4.4.3.1 Segment activation

A segment in the context model is activated whenever its pattern of pitches begins to match the pattern of pitches appearing in the input data. Specifically, if the current pitch interval in the input matches the first value stored in the context model segment, the segment is activated.

4.4.3.2 Agent instantiation

Upon activation of a segment, a *reactive listening agent* is instantiated from that segment. A listening agent is an autonomous entity that monitors the musical input and serves as a reactive embodiment of its *template* – the segment from which it was instantiated.

The goal of an agent is to advocate the interpretation of the musical input according to its template. It accomplishes this goal in two ways: prediction and parsing.

Listening agents are activated separately from the STM and LTM models. Agents are instantiated with a template, a frequency count, and a unique identification number. See Appendix B for a detailed description of the agent class declaration. There are times when agents with identical templates can appear in both the STM and LTM models, but since prediction and parsing are handled and recorded separately for STM and LTM, this is of no consequence.

4.4.3.3 Agent-based prediction

In the first phase of its short life, the *matching phase*, an agent continually compares its template with the input. If more than half of the template is matched correctly by the input, the listening agent begins to generate predictions based on the portion of its template still to be matched by the input. Due to the perceptually-based variable-order context models, appropriate multiple-step-ahead predictions are generated, as described in full detail in the next chapter. If a mismatch occurs at any point, the matching is halted and the agent terminates.

4.4.3.4 Agent-based parsing

If and when the input pattern matches the segment completely, the agent enters the *parsing phase*, during which it attempts to label the just-heard input pattern as being an instance of its template. A central contribution of this research is the multi-agent system developed to generate, maintain and reconcile multiple parsing hypotheses simultaneously and on-line. Parsing is described in full detail in Chapter 6.

4.4.3.5 Design Considerations

Maestro follows Minsky's idea that learning involves the addition of new agents [82]. Therefore, as a result of listening, a reactive model of the music is constructed.

In maintaining STM and LTM frequency counts, one option would have been to increment the frequency counts whenever a duplicate segment is suggested, similar to Conklin and Witten's system. However, unlike Conklin and Witten's system, the segmentation in Maestro is not exhaustive, and certain repetitions of a segment might be missed by the PGS segmentation modules. Therefore, in Maestro, frequency counts are incremented whenever an agent reports a complete match of the input with its template. This is a more targeted search for repetitions, as Maestro examines only those patterns originally delineated by perceptual cues, which according to the hypotheses tested in Chapter 9 are the more relevant patterns.

Another possible modification to Maestro's design is to immediately store each new segment as an agent, without maintaining a separate context model. This strategy is used by Rosenthal in [98], and the model he describes exists solely in the collection of rhythm agents. However, since it only maintains one instantiation of each agent, Rosenthal's system cannot track two overlapping occurrences of the same pattern. Maestro, on the other hand, maintains a separate context model from which listening agents can be instantiated as needed. This allows for the tracking of multiple overlapping instances of a single pattern.

4.5 Summary

This chapter has reviewed Maestro's modelling stage. When listening to music, people store information about what they hear for future use. This information may be stored in different types of memory, each having a different time-frame and function. Three types of musical memory are modelled in Maestro: echoic memory, short term memory, and long-term memory. Maestro follows the work of Conklin and Witten and stores melodic intervals in separate short and long-term context models.

The focus of this research is to make the context modelling approach more cognitively realistic. This is achieved primarily through the variable-order contexts that result from perceptually guided segmentation. This approach allows Maestro to generate multiple-step-ahead predictions. Additionally, Maestro's model stays within more cognitively realistic constraints on model size and growth, and results in more efficient models for prediction.

In keeping with the active listening approach, and with Jackendoff's theory of maintaining multiple simultaneous hypotheses when faced with ambiguity, Maestro incorporates the multi-agent system paradigm into its design. The context model is activated by instantiating various segments into autonomous reactive listening agents. Once instantiated, the listening agents predict and parse the musical information.

Chapter 5

Prediction

When listening to music, people generate predictions about what is to come next. This chapter describes Maestro's prediction capabilities, which include the generation of appropriate multiple-step-ahead predictions, and the handling of prediction ambiguity. First, the relevant background from cognitive musicology is reviewed. Then, entropy-based performance measurements are formally introduced. Finally, Maestro's prediction stage is described, with a focus on the agent-based prediction methodology.

5.1 Background

> *Music theorists and experimental psychologists will certainly agree that expectancy, both conscious and subconscious, plays a crucial role in the perception and cognition of music. (Narmour, [86, p. 417])*

The ability to generate expectations about upcoming events in a performance is a central aspect of music cognition. When listening to a piece of music, listeners use their previous musical experiences to guide their expectations about what is to come next [68]. This phenomenon is variously referred to as prediction, prospective hearing, expectation, implication, or anticipation. A review of the psychological literature about expectancy can be found in [110].

Musical prediction is closely related to musical modelling. Adachi and Carlsen cite experimental evidence that children internalise culture-specific melodic prototypes that, in turn, are claimed to be a basis of musical expectancy [1]. Krumhansl points out that studies of musical expectancy uncover a listener's knowledge about musical patterns and the psychological processes used to encode, organise, and remember music [68].

The expectations for what will happen at a particular time in the future can change as the piece progresses and more is learned about the music. Krumhansl therefore adds that studies of musical expectation offer insights into the dynamic processing of information over time, with continuously changing expectancies for subsequent events [68, p. 57].

Predictions are influenced by a number of factors – three levels of information used for making predictions are mentioned in the literature:

1. *Innate*: Narmour discusses bottom-up expectancies, resulting from the perceptually immediate musical context [86, 87]. According to Narmour, these expectancies are based on a universal set of innate melodic implications.

2. *Short Term Memory*: This level concerns information specific to a particular piece and is associated with Narmour's Intra-Opus style.

3. *Long Term Memory*: This final level concerns information abstracted from large number of sequences, and is associated with Narmour's Extra-Opus style.

As mentioned in Section 1.4.1, Maestro does not model innate musical preferences. Instead, the focus here is placed specifically on the learned aspects of musical intuition, and Maestro's design therefore addresses levels 2 and 3 above.

An additional distinction can be drawn concerning the type of schema used to generate expectations. The first type is a time-dependent event-hierarchy of a sequence of notes seen before and predicted to occur again. The second type is a time-independent tonal-hierarchy stipulating stability-conditions of certain pitches in a tonal framework (see [44, p. 133]). As Maestro does not maintain an explicit sense of tonality, only the first type of schema is handled.

Dowling states that expectancies may not always specify exactly what is to come next, but can instead be more general, allowing for small deviations [44, p. 167]. Similarly, Jackendoff [61] suggests that a specific musical expectation places constraints only on a certain set of musical parameters or features, leaving the others unconstrained. For example, a certain harmonic progression in a piece of Western music could create the expectation that the next chord be in the dominant scale degree. This constraint, however, leaves a degree of flexibility for the actual pitch of the soprano note, and does not in any way address the duration of the note or its loudness. With regard to these issues, Maestro's agents generate only specific predictions regarding pitch.

People's ability to generate musical predictions multiple-steps ahead of time was described in Section 2.2.3. As explained below, Maestro is capa-

ble of generating appropriate multiple-step-ahead predictions in accordance with the current musical context.

5.2 Performance Measure: Entropy

In Section 1.3.4, it was concluded that prediction performance is an objective measure for monitoring the progress of a system modelling music learning. The next question to address, then, is what method to use in measuring prediction performance. One option is to have the system output a single prediction about the upcoming note. This prediction can then be validated against the actual observed data and the ratio of correct to incorrect responses can be maintained.

While this method would reflect the progress of learning, it would not be an appropriate measure for music prediction, as Carlsen suggests that expectancies select a set of tones rather than a single tone [68, p. 58]. Sloboda also notes that an interesting melody has multiple, sometimes conflicting, implications [115, p. 163]. Therefore, a method is needed in which credit is appropriately assigned to these multiple predictions. For example, if a listener's second-ranked choice ends up being correct, the listener should not be evaluated as being totally wrong.

To this end, Conklin and Witten's music prediction system does not select only one prediction, but instead generates a full probability distribution, with probabilities assigned to all the different possible next-note values. The segments in the context model that match the current musical context are all used to generate predictions which are then integrated into one probability distribution. This integration is described below. The probability distribution is not a general purpose one, but rather one that is customised according to the specific musical context.

As a measure of performance, Conklin and Witten calculate the *entropy* in order to evaluate how well the generated probability distribution matches the actual observed note. Bishop discusses two interpretations of entropy [11, pp. 240-245]. One views entropy as information content, the other as a measure of disorder. These two types of entropy are used in Maestro for different purposes. They are now formally described in turn.

5.2.1 Information Content: Prediction Entropy

Entropy can be interpreted as a measure of information content, or degree of surprise. This type of entropy is referred to here as *prediction entropy*, and is the entropy used by Conklin and Witten in [132].

Prediction entropy measures the degree of surprise experienced by the system upon observing a specific event, and is given by the equation:

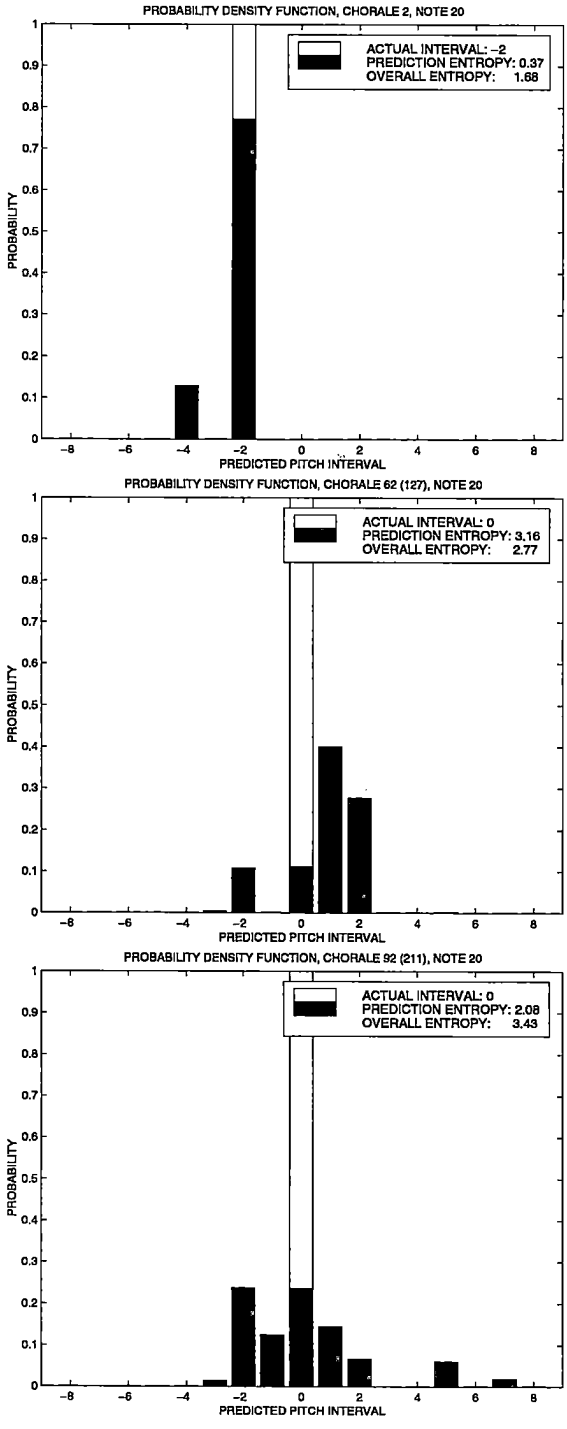


Figure 5.1: Various probability distributions generated by Maestro when predicting an upcoming pitch interval. The respective prediction entropy and overall entropy values for each probability distribution are shown.

$$E_p = -log_2(p_n) \tag{5.1}$$

where p_n is the probability assigned by the system to the event n occurring in the input.

The value of prediction entropy depends both on the probability distribution generated by the system and on the value that is actually observed. (Entropy is calculated here using the *hypothesis testing* approach [26, 125], in which the probabilities induced from the music heard so far (training set) are used to generate a probability distribution which is then combined with the incoming event (test set) in Equation 5.1.)

A totally certain event (probability of 1.0) leads to a prediction entropy of zero, while an unexpected event leads to a higher prediction entropy. Figure 5.1 shows examples of different prediction scenarios and the resulting entropy values when Maestro is predicting an upcoming pitch interval. The better the prediction, the higher the probability assigned to the actual observed value, and the lower the prediction entropy. Thus, the goal of a prediction system is to generate probability distributions that result in the lowest prediction entropy possible.

Prediction entropy is based on the landmark work in information theory by Claude Shannon [114]. In extending Hartley's earlier work, Shannon describes a scenario where a message is being sent over a noiseless communications channel. Both the sender and the receiver share a model of the set of probabilities of each message being sent. As a result of these probabilities, each message can be said to have a certain amount of information content: a high-probability message is expected by the receiver, and does not contribute much information, while a low probability message surprises the receiver, and is said to have a high information content. For efficiency in transmission, those messages appearing frequently (high probability) should be assigned shorter codes, while those appearing infrequently (low probability) should be assigned longer codes. Shannon's *noiseless coding theorem* states that the optimal length of a message in bits can be computed by the entropy formula given above.

5.2.2 Degree of Disorder: Overall Entropy

The second interpretation mentioned by Bishop views entropy as a measure of disorder and is referred to here as *overall entropy*. This measures the certainty of the prediction, or alternatively, the sharpness of the probability distribution generated by the system for the current musical context. Overall entropy is given by the equation:

$$E_o = -\sum_n p_n log_2(p_n) \tag{5.2}$$

where p_n is the probability assigned by the system to the event n occurring in the input, and the sum is calculated over all possible observable values n.

Unlike prediction entropy, overall entropy is independent of which note is actually observed; it depends only on the probability distribution generated by the system. A sharp distribution with all the probability mass placed on one value, receives an overall entropy of zero. Conversely, a relatively flat probability distribution, without any defined peaks (high uncertainty), receives a high overall entropy. Figure 5.1 presents some illustrative examples.

The highest overall entropy results from a totally flat distribution, with all possible values assigned equal probability (chance). This maximum overall entropy value depends on the number of possible observable values in the input. Conklin and Witten's system chooses between 20 possible pitches (middle C to G above the staff) [77], leading to a maximum overall entropy of 4.32. In order to maintain the flexibility to handle a larger range of pitch intervals occurring in different styles, Maestro chooses between 121 possible pitch intervals (-60 semitones to +60 semitones), leading to a significantly higher maximum overall entropy level of 6.92. The same maximum values for the two systems (4.32 and 6.92) also apply to prediction entropy.

5.2.3 The Zero Frequency Problem

While both types of entropy prove to be useful measures, an important complication arises: entropy is not defined for a probability of zero since the logarithm of zero is undefined. In the field of text compression, this is called the *zero-frequency problem* [7], which arises when a compression system using an adaptive model (based on Shannon's coding theory) encounters a value that it has not seen before. The model is unable to generate an appropriate code for the value since it has a probability of zero. To get around the zero-frequency problem, each possible value needs to be assigned a finite probability. This applies even if the system has no reason to believe from its experiences that a certain value will come next. In this way, the system "covers all the bases." Various ways of accomplishing this are described in [7, Ch. 3].

In addressing the zero-frequency problem and distributing some of the probability mass between all the possible values, a trade-off arises between maximising the probability assigned to correct predictions on the one hand, and minimising the damage of incorrect predictions on the other.

Conklin and Witten [27] deal with the zero frequency problem by arranging that every possible next event have some non-zero probability. This is accomplished by blending using the partial match blending algorithm [132] to address the zero frequency problem. In brief, this involves using lower-order context segments when higher order context segments have zero-frequency. Westhead and Smaill [125] use a similar method. Maestro approaches the zero-frequency problem slightly differently, as explained below.

5.2.4 Musical Entropy

Entropy and information theory have been used for many years to address a number of musical issues. Leonard Meyer [79], at least since the 1950's, has been a proponent of applying information theory to music. In the 1960's, Hiller and colleagues [53, 54] performed information theoretic analyses of musical scores. More recently in 1992, Coffman [24] used information theory to measure the originality of compositions produced by children before and after formal instruction. Westhead and Smaill [125] also use entropy for the purposes of style discrimination.

Following the work of Conklin and Witten [27], prediction entropy is used as an objective measure for gauging prediction performance, which depends in part on the system's model. As Conklin and Witten explain, the information content of each successive note in a piece of music is not an intrinsic musical property, but instead depends on the listener's own model of the musical style [132]. As a model improves, so do the predictions, thereby lowering the prediction entropy.

There is a limit to this improvement, however. Even expert human listeners, well-trained in a particular musical style, do not achieve 100 per cent (zero entropy) prediction. This is partly due to the notion that music contains intentional surprises to keep even the well-trained listener interested. As Minsky points out, music needs to strike a balance between "novelty" and "nonsense" [82].

Conklin and Witten [132] identify three types of musical entropy:

1. *Stylistic entropy* refers to the disorder inherent in a musical style.

2. *Perceptual entropy* refers to the disorder resulting from the listener's model, and therefore fluctuates from one listener to another.

3. *Designed entropy* is the intentional surprise inserted into a piece by a composer, over and above the baseline stylistic uncertainty.

While these are useful distinctions, Conklin and Witten caution that unfortunately, it is not possible to distinguish between these measures in

practice, and it is a matter of some debate whether the distinctions hold even in principle.

Conklin and Witten state that their work addresses only perceptual entropy - the level of surprise relative to the listener's model. While this research addresses perceptual entropy, it also studies stylistic entropy by performing experiments with music from different styles in Chapter 10. Additionally, designed entropy is measured by studying the inter-song variability within musical styles in Chapter 7.

5.2.5 Entropy as a Measure of Performance

In the present research, entropy is used as a measure of prediction performance. The next question then becomes: What is considered to be a "good" level of prediction performance?

As Conklin and Witten explain, abstract measurements hold little meaning if one does not know how well people perform the same task [132]. To this end, Manzara *et al.* [77] conducted extensive experiments with human listeners to obtain a set of reference entropy values. They found their machine model performed almost as well as the human subjects in predicting Bach chorale melodies. More importantly, their machine-based results were generally correlated with the human results - the model did well where humans did well, and made mistakes where humans made mistakes. Chapter 7 presents the results of experiments conducted to compare Maestro's machine output with the human measurements collected by Manzara.

While the *absolute* measurements of machine entropy are of little value without the reference numbers of human performance levels, such human data is difficult to obtain, and it is outside the scope of this research to collect further human data. However, large sets of machine-generated prediction entropy results are able to capture the gradual improvement resulting from learning with experience. Such a *relative* measure of improvement is well suited to Maestro's focus on studying the process of music learning, and is used in analysing the experiments performed over the course of this research.

5.3 Prediction in Maestro

Predictions in Maestro are generated by the individual listening agents. Maestro is able to generate appropriate multiple-step-ahead predictions as it receives each input event, and due to its on-line approach, Maestro can report its changing perceptions and expectations as each note is received.

5.3.1 Agent-Based Prediction

Minsky describes the method of *partial recognition*, whereby, having already occurred in the past, a certain pattern can be re-detected as it begins to

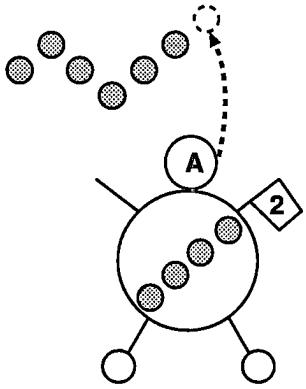


Figure 5.2: Agent-Based Prediction. Agent A is shown having a three interval (four note) pattern and a frequency of two. Upon matching the first two intervals of its template with the input, the agent generates a prediction that the final interval in its template will appear next.

appear again, and an expectation is generated suggesting that the pattern will appear in full [82]. This method has been implemented by Rowe [102] and Conklin and Witten [132], and is also used in the present research.

Predictions in Maestro are generated by the listening agents active in the system at the time. When an agent notices that at least half of its template has been matched by the recent musical input, it generates a prediction in accordance with the remaining portion of its template pattern, yet to appear in the input (Figure 5.2). No prediction takes place for the first note of a piece, as there is no context from which to generate any predictions.

As described in Section 2.2.3, depending on the specific musical context, people are able to generate predictions more than one step ahead. Maestro predicts as far ahead as its current context allows. The number of steps ahead, or forecast horizon, of each prediction depends on the listening agent's template. For example, the agent with template $ABCDE$, after seeing ABC, will generate the prediction DE (2 steps ahead of time).

The length of the prediction is thus determined by the length of the original segment stored in the model. The optimal length of the context used for prediction (the order of the model) can vary with the particular data being examined. It is argued that by storing segments of various lengths according to the perceptual cues in the musical surface, Maestro predicts as many steps ahead as is appropriate for its currently activated musical context. This is in keeping with Maestro's focus on flexibility and learning.

Conklin and Witten perform their tests only on predictions made one step in advance [27]. Experiments in multiple-step prediction reported in Chapter 7 show that it is worthwhile to store segments of various lengths for

the purposes of generating longer-horizon predictions. The experiments also study the effects of training and forecast horizon on prediction performance.

5.3.2 Prediction Ambiguity

Prediction ambiguity can arise when multiple contexts match the input, causing multiple agents to be activated. Simultaneous, often conflicting, predictions can then be generated. In Maestro, the different agents' predictions are integrated into an aggregate probability distribution, and weighted according to two factors:

1. Reliability – The number of times this pattern has occurred before.

2. Certainty – The length of the context matched by the input.

The first factor is analogous to Conklin's and Witten's weighting described in [27], and is based on the idea that patterns that have appeared often in the past are more likely to appear again in the future.

The second factor has a basis in Bell *et al.* [7, p. 143], where once a model is trained, higher order models are given more emphasis than lower order models. This relies on the concept that matching a longer musical context is less likely to happen by chance than matching a shorter one. Therefore, predictions based on longer musical contexts are likely to be more accurate than those based on shorter contexts. The certainty factor is similar in function to the λ weights described by Ponsford *et al.* [91].

The total weight of a prediction is given by the pattern's frequency of occurrence multiplied by the length of context matched so far: reliability multiplied by certainty. When comparing predictions, weights for each value predicted are accumulated such that multiple identical predictions, each with a low weight, can combine to a greater overall probability than a single prediction with a high weight. For example, two agents both independently predicting A with weight 0.4 (total score 0.8) will result in a higher probability than one agent predicting B with score 0.6.

Maestro integrates its various predictions into an aggregate discrete probability density function (PDF). To deal with the zero-frequency problem, Maestro blends this PDF with a flat distribution, weighting the two by 0.9 and 0.1 respectively. From this final blended PDF, overall entropy is calculated as a measure of how certain Maestro is of its prediction. Once the actual value is observed by the system, the prediction entropy is calculated as a measure of the resulting level of surprise.

Predictions are integrated independently for the various prediction horizons. Predictions generated by STM and LTM agents are integrated separately. Similarly, in measuring and reporting prediction performance, Maestro keeps track of STM and LTM results separately. While Conklin and Witten combine the STM and LTM models to achieve optimal prediction results, the focus here is on studying the individual components of music learning. This research focuses primarily on long-range music learning, which has its basis in LTM. Therefore, most of the results reported in this dissertation rely on the LTM context model, unless otherwise indicated.

Maestro's prediction performance depends on the quality of its model. As the model develops with experience, Maestro's predictions improve, lowering the prediction entropy.

Apart from measuring prediction performance, entropy can also serve as a good measure of perceptual complexity and cognitive load, which may increase when handling ambiguous passages. Conklin and Witten state that meaningful comparisons between individual pieces or styles can be undertaken by objectively measuring perceptual complexity in terms of entropy [132]. Experiments reported in Chapter 8 are conducted to study prediction ambiguity. Three different measurements (prediction entropy, overall entropy, and a measure of agent activation) are used to identify and study different types of prediction ambiguity.

5.4 Summary

This chapter has reviewed Maestro's prediction capabilities. When listening to a piece of music, people generate expectations about what is to come next. These expectations change over time and are determined both by the performance itself and by the listener's internal model.

Prediction in Maestro is performed by the individual listening agents. Agents generate predictions using the partial recognition strategy. Previous studies, such as Conklin and Witten's, focused only on predictions made one-step ahead. Due to the variable order contexts in Maestro's model, the listening agents generate appropriate multiple-step-ahead predictions, according to the current musical context.

The predictions of the various agents are integrated into a probability density function (PDF) and weighted according to the factors of certainty and reliability. This PDF is combined with a flat distribution in order to address the zero-frequency problem.

Whereas Conklin and Witten use one type of entropy to measure prediction performance, two types of entropy are used in Maestro: overall entropy indicates the flatness of probability distributions generated and thus the certainty of the predictions, while prediction entropy reflects the degree of surprise experienced when observing certain events.

Chapter 6

Parsing

Listening is an iterative process through which a person assembles a developing understanding of a piece of music. Maestro communicates its interpretation of a musical performance through a process called *parsing*. Parsing involves labelling a performance in accordance with the structural interpretation suggested by Maestro's model.

This chapter first presents some background on musical parsing, drawing from the closely related field of Natural Language Processing. Various machine models of musical parsing are then considered. Maestro's approach to parsing, which incorporates a distributed agent-based parsing algorithm capable of handling parsing ambiguity, is described. Finally, issues of cognitive load and musical tension are discussed, and the circularity problem introduced in Section 3.2.6 is addressed in full.

6.1 Background: Musical Parsing

> *A piece of music is a mentally constructed entity ... the central task of music theory should be to explicate this mentally produced organisation. (Lerdahl and Jackendoff, [72, p. 2])*

Like musical prediction, musical parsing depends on the listener's stored model of musical experience. Narmour states that musical understanding occurs when the top-down schemata stored in a listener's model match the bottom-up perceptual information emerging from the musical surface [86, p. 11].

Similar to other aspects of music listening, parsing is an online process and can only make use of the information heard so far in a piece. The on-line nature of parsing can lead to complications in situations of ambiguity since there is often insufficient information available to decide between multiple parsing interpretations. This problem is noted both by Jackendoff [61] and

Berent and Perfetti [8]. Jackendoff explains that at a later point in the piece enough information may become available to resolve the ambiguity and the listener can go back and reinterpret the music, thus resulting in the effect of retrospective listening described in Section 2.2.6.

Music parsing poses a number of significant challenges to machine modelling. It depends heavily on the listener's internal model, is performed on-line, and often involves dealing with ambiguity and performing retrospective listening. Maestro's parsing stage addresses all these issues. Before discussing Maestro's design, it is appropriate to review a closely related and well developed field – Natural Language Processing, and examine how the issues addressed relate to Maestro's handling of musical parsing.

6.1.1 Music and NLP

Natural Language Processing (NLP) research focuses on improving the ability of computers to handle human language and grammar. As there are many similarities between the human perception of language and the human perception of music [91], many researchers have developed grammatical models of musical understanding. Jackendoff states that the perception of music parallels the perception of language; it involves the unconscious construction of abstract musical structures, of which the events of the musical surface are the only audible part. These abstract musical structures are what account for one's musical understanding and are what make listening to music more than just hearing a sequence of pitch events [61].

This section will first review some relevant aspects of human linguistic processing and parsing, and relate them to musical concepts. Then, drawing from the NLP literature, different issues in designing machine parsers will be presented in light of these considerations.

6.1.1.1 Human Language Parsing

It is widely believed that people have an internal grammar which they use to understand and generate language [3]. People can use a grammar to determine whether or not a spoken sentence is grammatical. This task is called recognition. For example, consider the two sentences:

```
I own a car.
I car a own.
```

The first sentence is grammatical while the second is obviously not. Gazdar and Mellish point out that while determining the grammaticity of a sentence is an important and challenging task, it is completely dwarfed in practice by a much more serious problem, that of pervasive natural language ambiguity [50, p. 169].

Ambiguity When parsing sentences of natural language, people must deal with various types of ambiguity. Gazdar and Mellish mention three types of ambiguity [50, pp. 169-177], referred to here as Lexical, Structural, and Parsing Segmentation ambiguity:

1. Lexical Ambiguity – The meaning of a word or group of words is unclear.

   ```
   I walked to the bank.
   ```

   `bank` can refer to a monetary institution or to a river's edge. In either case, however, it is clear that it indicates the destination to which the subject has walked.

2. Structural Ambiguity – The function of a certain word or group of words in the sentence is unclear.

   ```
   I observed the boy with the telescope.
   ```

   Either the boy is described as having a telescope, or the telescope was used to observe the boy. The meaning of the individual words is clear, but their function is not.

   Both lexical and structural ambiguity can also appear together in one sentence.

   ```
   I saw her duck.
   ```

   Lexically, the meaning could be either 'I saw her bend down', or 'I saw the bird belonging to her'. Structurally, the direct object of `saw` could be `her` or `duck` respectively.

3. Parsing Segmentation Ambiguity – In processing written text, separate words are delineated by spaces. However, in natural spoken speech, determining the word boundaries is sometimes difficult [50], as in:

   ```
   Madam, I'm Adam Ant.
   Madam, I'm adamant.
   ```

   In the spoken speech signal, it may be unclear which of the above two sentences is being spoken. People can rely on previous experience in resolving this type of ambiguity [69].

Having outlined three types of ambiguity, it now remains to determine which, if any, are also relevant in the musical domain. As there is no widely-agreed upon method of assigning formal meaning in music, lexical ambiguity

does not readily apply in the musical realm. In grammar-based systems that construct hierarchical structures from music, structural ambiguity is relevant. However, Maestro does not do this.

Unlike the first two types, parsing segmentation ambiguity is relevant to Maestro's handling of music. Russell and Norvig [105] state that the first stage of a text understanding system is tokenisation – breaking the input up into words, or atoms for the system to deal with. This is a difficult task in music because there is ambiguity as to which notes should be grouped together to form the atomic musical segments. Parsing segmentation ambiguity is dealt with by Maestro's parsing stage.

It is important to clarify the distinction between two closely related types of ambiguity addressed by Maestro: the segmentation ambiguity described in Chapter 3 and the parsing segmentation ambiguity described here:

1. *Segmentation Ambiguity* – This exists in the context of discontinuity-based segmentation (segmentation), and arises when various perceptual cues suggest conflicting ways of segmenting the data.

2. *Parsing Segmentation Ambiguity* – This exists in the context of repetition-based segmentation (parsing), and arises when the repetitions of two previously seen patterns overlap with one another in the music.

Maestro's handling of the former was described in Chapter 3. The latter is addressed below in Section 6.2.

Parsing ambiguity poses additional complications in cases where a decision needs to be made before sufficient information is available. This is known as deterministic parsing.

Human Deterministic Processing Human parsing is *deterministic* [3, 61]. People use the information present at the time to arrive at the best possible understanding of a sentence. In ambiguous situations, people may entertain multiple simultaneous interpretations for a short while. However, if the ambiguity persists, one of the interpretations must be chosen before proceeding on to the rest of the sentence.

A common result of human deterministic parsing is the *garden path* effect. Certain sentences are said to mislead people on a first reading, as if 'leading them down the garden path.' The result is a breakdown in parsing, and the person must then start over again from the beginning:

```
The horse raced past the barn fell.
The prime number few.
The granite rocks during the earthquake.
```

People usually start off understanding the first sentence to mean that the horse ran past the barn, only to find that the word `fell` cannot be made to fit with this reading. They then go back and read the sentence again, this time understanding that:

```
The horse, [that was] raced past the barn, fell.
```

Had the person simultaneously maintained both interpretations of the word `raced` until the end of the sentence, the word `fell` could have been incorporated successfully into one of them. However, since the alternate meaning of the word was pruned away at some stage, parsing broke down at the end of the sentence, and parsing had to be restarted from the beginning. Jackendoff cites this as evidence that the human parser is deterministic [61].

Deterministic parsing means that people must choose one of the possible interpretations, even though there is insufficient information to decide the matter conclusively. People must therefore employ certain selection factors or heuristics to choose between the various interpretations and resolve the ambiguity.

Selection Factors Allen [3, pp. 161-164] discusses certain general principles concerning the way people resolve ambiguities:

1. Minimal Attachment – People prefer the understanding with the simplest structure, namely the one that leads to least complex or convoluted parse tree. For example:

   ```
   We painted all the walls with cracks.
   ```

   Against common sense, minimal attachment results in a tendency to read this sentence as meaning that the cracks were painted on to the walls, or that the cracks were used as an instrument to paint the walls. The intended meaning is that the walls with cracks on them were painted.

2. Right Association or Late Closure - New constituents tend to be interpreted as part of the current constituent under construction:

   ```
   George said that Henry left in his car.
   ```

   While two meanings are technically possible, most people understand that Henry left in his car rather than that George spoke in the car.

   Sometimes the above two principles are in conflict with each other.

   ```
   The man kept the dog in the house.
   ```

The sentence could either be answering the question: "Which dog did the man keep?", or the question "where did the man keep the dog?" Right association prefers the former reading while minimal attachment prefers the latter. A third principle is needed to decide between them:

3. Lexical Preference – Certain words are more commonly used in some settings than others.

```
I chose the dog in the house.
I put the dog in the house.
```

Depending on the verb used, the preferred meaning changes. This method of resolving the ambiguity relies on a formal framework for representing this lexical knowledge. This knowledge can encode a set of probabilities indicating which readings are more likely in various contexts.

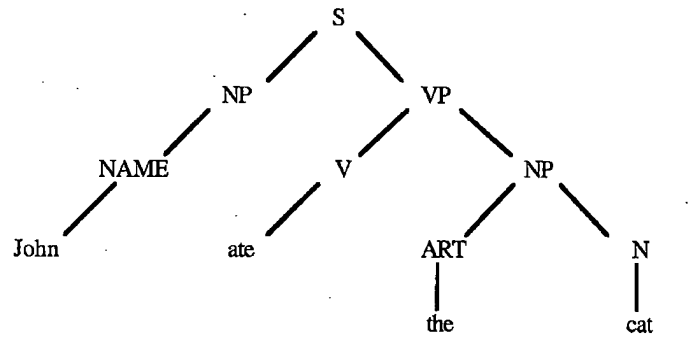
How does each of the principles used to resolve ambiguity relate to Maestro's handling of music? In grammar-based systems that construct hierarchical representations from the music, minimal attachment would be relevant, and would prefer the least complex structures. As stated above, Maestro does not deal with hierarchies, so this is not relevant to the present research.

Right association is relevant to Maestro's parsing of music. Maestro's parsing stage prefers longer patterns to shorter ones.

The concept of lexical preferences does not directly apply to the musical realm, as there is no clear framework for assigning formal meaning in music [120]. However, musical parsing is compatible with the general strategy of maintaining a set of probabilities to indicate which readings are more likely. This approach is embodied in Maestro's context model, and is used in the parsing stage, where patterns occurring more frequently are preferred to those occurring less frequently.

Thus, both right association and the probabilistic aspect of lexical preferences are used by Maestro in resolving parsing ambiguity, as described in more detail in Section 6.2 below.

So far, this chapter has covered different types of ambiguity and the human preferences in dealing with ambiguity deterministically. These issues are now addressed in the context of designing a machine model of parsing.



1. S → NP VP
2. VP → V NP
3. NP → NAME
4. NP → ART N
5. NAME → John
6. V → ate
7. ART → the
8. N → cat

Figure 6.1: Parsing an example sentence `John ate the cat` (top) using a simple grammar (bottom). From (Allen, [3]).

6.1.1.2 Issues in Machine Parsing

According to Gazdar and Mellish, parsing involves interpreting the input through computing the structures assigned to a given phrase by a given grammar [50].

A grammar is a set of rules or *transformations* that indicate which syntactic forms are permissible in constructing sentences. When presented with a sentence, a system can parse it by attempting to label the various words or groups of words using the rules in the grammar. A simple grammar and parsing example are given in Figure 6.1.

A number of issues need to be addressed when designing a machine parser, including:

- left-to-right versus right-to-left;
- top-down versus bottom-up;
- breadth-first versus depth-first;
- partial versus complete parsing.

Each of these is now discussed in turn.

Left-to-right versus Right-to-left A parser can proceed through the input from left to right or from right to left. Alternatively, it can be permitted

random access to the input, allowing the parser more flexibility in performing its task. However, as Gazdar and Mellish note, parsers that attempt to parse a natural language as it is being produced, whether as it is spoken or as it is typed, are forced to adopt a strategy that is basically left-to-right [50, p. 151]. As music is also processed as it is being produced, Maestro, a cognitively realistic musical parser, must process the input left-to-right.

Bottom-up versus Top-Down In attempting to construct appropriately structured hierarchical parsing trees from the input, a parser may proceed in two ways.

Top-down parsing starts off with a set of possible goals (high level structures) and tries to fill them with the given words. For example, a parser might start out with the high level description of a sentence '`NOUN-PHRASE VERB-PHRASE`', and then attempt to fit this general form on the words appearing in the sentence. If this high level form does not work, another may be tried. The top-down approach may be analogous to an experienced human listener who tries to use common and familiar forms to parse the music, and is similar to the time-span and prolongational reductions described by Lerdahl and Jackendoff [72].

In contrast, bottom-up parsing begins with the words appearing in the input, and various possible higher-level forms that fit with the input are constructed. The parser then searches through the space of alternatives to try and build a consistent high-level understanding of the sentence. Not all of the possibilities lead to a solution and it is also possible to arrive at multiple solutions, giving rise to parsing ambiguity.

As Maestro does not explicitly deal with hierarchical structures, an effectively bottom-up approach is chosen for Maestro's parsing stage.

Breadth First versus Depth-First A comprehensive parser often needs to investigate many possible interpretations, all of which can be represented in a search tree. The parser can either search this tree breadth-first or depth-first. Depth-first search entails following through with a single interpretation completely before proceeding to the next one. In contrast, a breadth-first strategy involves maintaining a number of hypotheses simultaneously, advancing each in turn by a single step.

While a depth-first approach uses less memory and requires less bookkeeping than a breadth-first strategy, it suffers from a pervasive need for backtracking. As seen from the parsing breakdown experienced when parsing garden path sentences, people do not seem to be able to backtrack significantly. Therefore, a breadth-first strategy is chosen in Maestro.

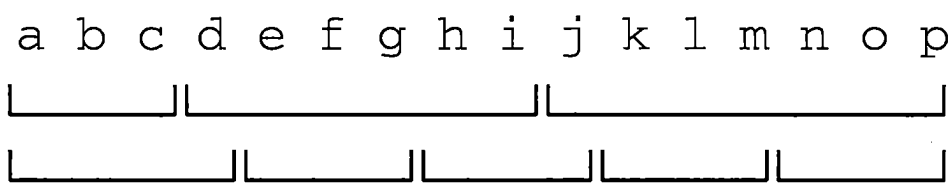


Figure 6.2: An example of the drawbacks of greedy parsing. A greedy algorithm would choose the bottom parsing interpretation even though the top one is better overall.

Partial versus Complete Parsing Usually, the goal of a parser is to find a grammatical expression spanning an entire sentence. However, this is not always possible. It may be that a grammar is not developed enough to handle the input, or that the input itself is simply not reducible to one, overall complete parse.

Under the complete parsing approach, a sentence is declared ungrammatical unless a full spanning solution is found. On the other hand, partial parsing attempts to build up solutions for different parts of the input, and integrates them to whatever extent possible.

Lerdahl and Jackendoff discuss partial parsing in the framework of music and state that a fundamental feature of musical understanding is that it functions simultaneously in varying degrees of fragmentation and integration. An undeveloped or inattentive listener might hear only partial structures, either unconnected or insufficiently integrated with each other [72, p. 197].

Especially in the context of music learning, where a grammar may not be fully developed, the partial parsing approach is more appropriate, and is thus chosen in this research.

6.1.2 Machine Handling of Parsing Ambiguity

As mentioned above, parsing often involves dealing with ambiguity. In designing a machine parser, the issue of ambiguity handling must be addressed.

6.1.2.1 Greedy versus Optimal

In parsing ambiguous input, Bell *et al.* draw a distinction between two strategies: greedy and optimal [7]. Greedy parsing proceeds from left to right. Whenever it is faced with an ambiguous situation, in keeping with the principle of right association, a greedy parser chooses the longest segment possible for the immediate context. However, this may lead to inefficiencies, as shown in Figure 6.2, where the greedy strategy ends up choosing a solution with

many short segments. The better solution is rejected because it happens to begin with a shorter segment.

As the name suggests, an optimal parser attempts to find the best overall solution. In order to find the optimal solution, intermediate results are stored by the parser when processing the data, before deciding on a final interpretation.

6.1.2.2 Storing Intermediate Results

When going through the search tree, a parser can keep track of what it has discovered in order to avoid rechecking those options that lead to failure. Storing intermediate results also allows for the maintenance of multiple simultaneous interpretation hypotheses. Gazdar and Mellish [50, Ch. 6] discuss two strategies for storing intermediate information: Well Formed Substring Tables (WFSTs) and Charts.

WFSTs will not be discussed here in detail, and the interested reader is referred to Chapter 6 of Gazdar and Mellish [50]. WFSTs can store partial analyses and thus help a parser to avoid rechecking specific sections of the input. However, since various larger scale parsing hypotheses and goals are not encoded into the WFSTs, there is nothing to stop a parser from re-investigating certain higher-level hypotheses it has already attempted before. In order to represent parsing goals, another method of storing intermediate results is necessary.

6.1.2.3 Chart Parsing

Charts are data structures that capture both parsing goals and structural information. Charts contain arcs called edges. Each edge is decorated with a grammar rule, and a 'dot' indicating how much of the rule has been matched by the input.

A chart parser proceeds through the input and examines different ways of grouping words in an attempt to come up with a parsing solution [3]. A chart parser uses a chart to keep track of intermediate results, thereby avoiding the need for backtracking as well as avoiding the need to re-investigate options it has already examined. Figure 6.3 shows an example of chart parsing.

Edges can be either active or inactive. Inactive edges are those which have been completely matched by the input. Active edges are those that have not been completely matched by the input, and represent a parsing hypothesis that still needs to be confirmed or rejected with respect to the input.

When the first part of a rule in the grammar matches the input, an edge is added to the chart. The key strategy of a chart parser is to find which inactive edges can be used to complete the active edges. Thus, large scale, hierarchical parsing structures are constructed. Chart structures are completely neutral with respect to parsing strategies. In left-to-right chart parsing, simultaneous on-line parsing hypotheses are maintained.

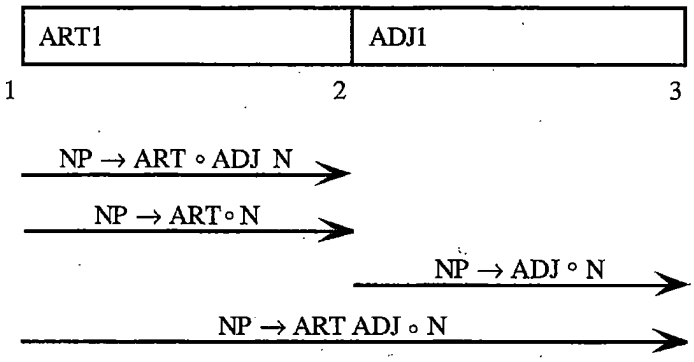


Figure 6.3: An example of Chart Parsing from (Allen, [3]). The input is shown at top labelled with certain parts of speech. The rules of the grammar are shown at the bottom. The edges (all active) generated from the input and grammar by the chart parser are shown in the middle.

6.1.2.4 Deterministic Approaches

As mentioned above, human parsing is deterministic. Therefore, a machine model of human parsing should also be deterministic. Gazdar and Mellish explain that from an engineering perspective, one could design a non-deterministic parser that never breaks down on well-formed strings [50, p. 175]. Thus an approach less concerned with cognitive realism can achieve greater levels of optimisation. However, for a more cognitively realistic, deterministic system, sentences that people tend to misparse will be misparsed by the system [3, p. 159].

When faced with persistent ambiguity, a deterministic parser must often make decisions based on insufficient information. In order to resolve the ambiguity, certain selection criteria or heuristics must be employed. The discussion above mentioned three principles used by humans in deterministically resolving natural language ambiguity: minimal attachment, right association, and lexical preferences.

Gazdar and Mellish suggest that a deterministic parser can use heuristics

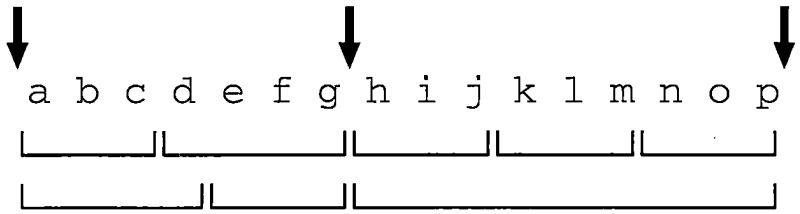


Figure 6.4: An example of cut-points encountered in parsing, indicated by the arrows. Ambiguities are contained within the regions lying between the cut-points.

based on the relative frequencies of different constructions [50, p. 175]. In a similar vein, Allen discusses the use of estimated probabilities of different constructions, derived from a large enough sample of data. This use of probabilities is related to the lexical preference principle described above. Maestro's heuristics for disambiguation include right association (longer patterns preferred), and the use of a set of probabilities similar to lexical preference (more frequent patterns preferred).

Once a set of heuristics is chosen, there still remains the issue of when disambiguation should be forced – how long must an ambiguity persist before the parser resolves it, despite having insufficient information?

One option would be to set a fixed limit, either in terms of time, or number of events [8]. Another option commonly used in parsers is to rely on *cut-points*. Cut-points are locations in the input stream where there are no overlapping parsing possibilities. Even though ambiguities may persist in regions lying between two cut-points, they are contained within those regions, and their resolution is independent of ambiguities elsewhere (Figure 6.4).

Jackendoff cites experimental evidence that multiple syntactic analyses are computed in parallel by humans, and that the clause boundary is the point at which all but the most salient analyses are discarded [61]. In this way, a clause boundary serves as the cut-point in language processing. The concept of cut-points is also relevant in the context of musical parsing, as Lerdahl (cited by Jackendoff in [61]) suggests that cadenced group boundaries are points where less stable analyses are pruned.

Maestro follows this approach and resolves parsing ambiguities at cut-points. The distributed parsing algorithm described below results in multiple hypotheses being maintained within the regions specified by cut-points.

So far the discussion of Natural Language Processing has yielded a number of interesting issues relevant to Maestro's handling of musical parsing. First, parsing segmentation ambiguity seems to be most relevant for this

research. Second, parsing is performed on line, and ambiguity must often be resolved based on certain heuristics. Third, a resulting delay in processing can lead to retrospective listening. Fourth and finally, the deterministic heuristics of right association and lexical preference are relevant to Maestro's parsing stage. The next section will deal with machine modelling of musical parsing.

6.1.3 Machine Models of Musical Parsing

Berent and Perfetti note the importance of the real-time processes involved in the on-line parsing of music to developing an accurate psychological account of the listening process [8]. Jackendoff notes that the original Generative Theory of Tonal Music published in 1983, does not account for the ways in which various musical structures are assigned to the musical surface on-line. In his 1991 paper *Musical Parsing and Musical Affect* [61], Jackendoff addresses the question of on-line musical parsing. In addressing the issue, Jackendoff considers three types of parsers: serial single choice, serial indeterministic, and parallel multiple analysis. Each is now described in turn.

Serial Single Choice Parser The first parser considered by Jackendoff is a serial single choice parser. When facing ambiguity, the parser chooses only one possibility – the one it considers to be most likely. If this choice proves to be wrong, the parser backtracks and chooses another. Jackendoff notes a problem with this approach – the parser would spend an unrealistic amount of time backtracking while, in the meantime, new music is streaming in. As backtracking is time consuming, the processing load imposed by the music must be borne by increasing speed of the parser. Therefore, Jackendoff rejects this parsing model.

Serial Indeterministic The second option considered by Jackendoff is a serial indeterministic parser. This approach delays decision until enough information is available. The parser computes preliminary analyses of local parts of the input, but does not integrate them into a global analysis until a single correct structure can be determined for the whole selection. The benefit of this delayed analysis approach is that it avoids the onerous backtracking that plagues the serial single choice parser. However, Jackendoff notes that this extended delay in parsing is not realistic. For example, people clearly have metrical intuitions long before the definitive evidence arrives [61].

Parallel Multiple Analysis The third and final parsing model considered by Jackendoff is the parallel multiple analysis model. According to

this approach, when the parser encounters ambiguity, processing splits into multiple simultaneous branches to represent the various interpretation possibilities. When a particular branch drops below some threshold of plausibility, it is abandoned. Whichever branches remain at the end of a piece then contain viable structures for the piece as a whole [61, p. 213].

Jackendoff cites evidence from linguistics research that multiple meanings are active immediately after ambiguous words are heard; many words have multiple meanings, yet the listener must select the correct meaning for the context of the sentence in which the word occurs. Jackendoff suggests that in cases of unresolved ambiguity, multiple analyses are maintained for a period of time, but are eventually pruned to a single sense, usually the more common one – even if still ambiguous from the input itself. In this way, the parser is deterministic.

Jackendoff presents a theory of on-line musical parsing based on linguistics and cognitive musicology. While he proposes a general approach, he does not give many technical specifics, nor is an implementation carried out. Maestro's parsing stage is designed to fill this void, and is an implementation of certain key aspects of Jackendoff's theory.

6.2 Parsing in Maestro

The present research is concerned with building a basic working implementation of Jackendoff's musical parsing theory and testing it with musical input.

6.2.1 Agent-Based Parsing

In Maestro, parsing is performed in a distributed fashion by the individual listening agents. A listening agent, upon successful matching of its entire template with the musical input, labels the appropriate part of the musical input with its unique identifier. This is called making a *parsing attempt* (Figure 6.5), and by doing so, the agent reports its interpretation of the input – how it believes the music should be parsed.

The overall goal of parsing is to formulate a consistent parsing interpretation of the performance, in which none of the labelled segments overlap with one another. However, the templates of different agents often overlap in the musical input; certain notes may fall into a number of different agents' parsing attempts. This leads to parsing ambiguity, more specifically to the parsing segmentation ambiguity discussed above.

Parsing segmentation ambiguity may occur either because patterns stored in the model happen to be similar, or because multiple overlapping segmen-

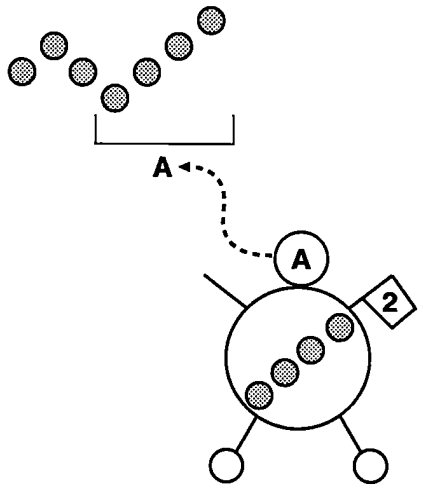


Figure 6.5: Agent-based parsing. Upon successfully matching its entire template with the input, the agent labels the occurrence of the pattern with its label.

tation hypotheses of the performance were originally stored in the model. Whatever the reason, such parsing ambiguity arises frequently, and in keeping with Maestro's design principles and Jackendoff's theory, multiple interpretation hypotheses are generated and maintained in the form of listening agents.

To reconcile competing parsing interpretations, Maestro takes advantage of the inherent suitability of multi agent systems for carrying out competitions between various agents. The specifics of the parsing competition are described below. Agents can compete and cooperate with each other, as each pursues its own goals. From the simple actions and interactions of the various agents, the complex parsing behaviour of the system as a whole emerges, yielding a unified, consistent parsing of the performance.

6.2.2 Parsing Competition

The C++ class description for a Maestro listening agent is shown in Appendix B. The functions included there are used to implement the parsing competition described in this section.

Agent based parsing consists of three main steps:

- Instantiation;
- Matching phase;
- Parsing phase;

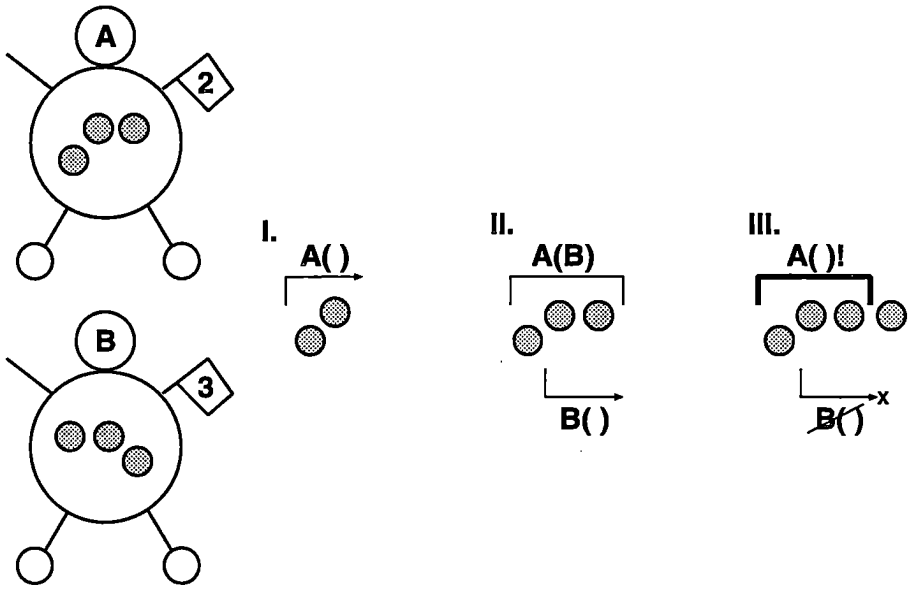


Figure 6.6: A parsing competition between two agents, A and B. I, II and III show the progressing steps of the competition. A is threatened by B (B has a higher frequency) and so A adds B to its list of threatening competitors, indicated by the () symbols. When B encounters a mismatch between its template and the input, A wins the competition, as indicated by the ! symbol.

6.2.2.1 Instantiation

In order to resolve ambiguity, selection factors are needed. To this end, in Maestro, one listening agent is said to be *preferred* to another. Preference is determined at the time of instantiation by which agent has a longer pattern (Right Association), or in the case of equal pattern length, which has a higher frequency (Lexical Preference). If both these characteristics are equal between the two agents, the agent with the earlier instantiation time is preferred. In this way, preference is a universally agreed-upon relation. If two agents overlap, the more preferred one is said to *threaten* the less preferred one. As soon as an agent is instantiated, it starts to maintain two lists. The first list contains those agents who threaten it, and the second list contains those agents whom it threatens.

6.2.2.2 Matching Phase

Once instantiated, a listening agent's life cycle consists of two parts: the matching phase and the parsing phase. During the matching phase, the agent compares its template against the musical input and generates predic-

tions as appropriate. If at any point there is a mismatch between template and input, the agent terminates. As it terminates, it notifies all agents whom it threatens that it no longer poses a threat to them so that they can stop tracking it.

6.2.2.3 Parsing Phase

If an agent succeeds in matching its template fully against the musical input, it enters the second phase of life – parsing. The agent's sole goal now becomes to have its parsing attempt chosen as that of the system as a whole. This is only possible if the agent does not have any other agents threatening it. In such a case, the agent notices that its first list is empty and thus proceeds to win the parsing competition. This is called *winning a parsing attempt.* The winning agent tells the agents threatened by it that it has won, and that they should all terminate.

A threatening competitor can be removed in two ways: either the threatening competitor encounters a mismatch between its template and the input, or a third agent beats the threatening competitor, causing it to terminate. An example of each scenario is now given:

1. Figure 6.6 shows two agents A and B.

   - (I) A is instantiated when the pattern it represents begins to appear in the music.
   - (II) A notices that it is being threatened by agent B who has a higher frequency and therefore adds B to its list. A does not immediately give up and terminate, as there is still the chance that B's template will not fully match with the musical input, and B will terminate on its own, thus ceasing to pose a threat to A.
   - (III) B indeed terminates prematurely upon finding a mismatch, and agent A removes B from its list. A finds that its list is empty and wins the parsing attempt.

2. Apart from a competitor failing to completely match its template to the input, threats may be removed in a more complex way. Figure 6.7 shows three agents A, B and C.

   - (I) A is instantiated when the pattern it represents begins to appear in the music.
   - (II) A notices that it is being threatened by agent B (higher frequency and longer template) and adds B to its list.

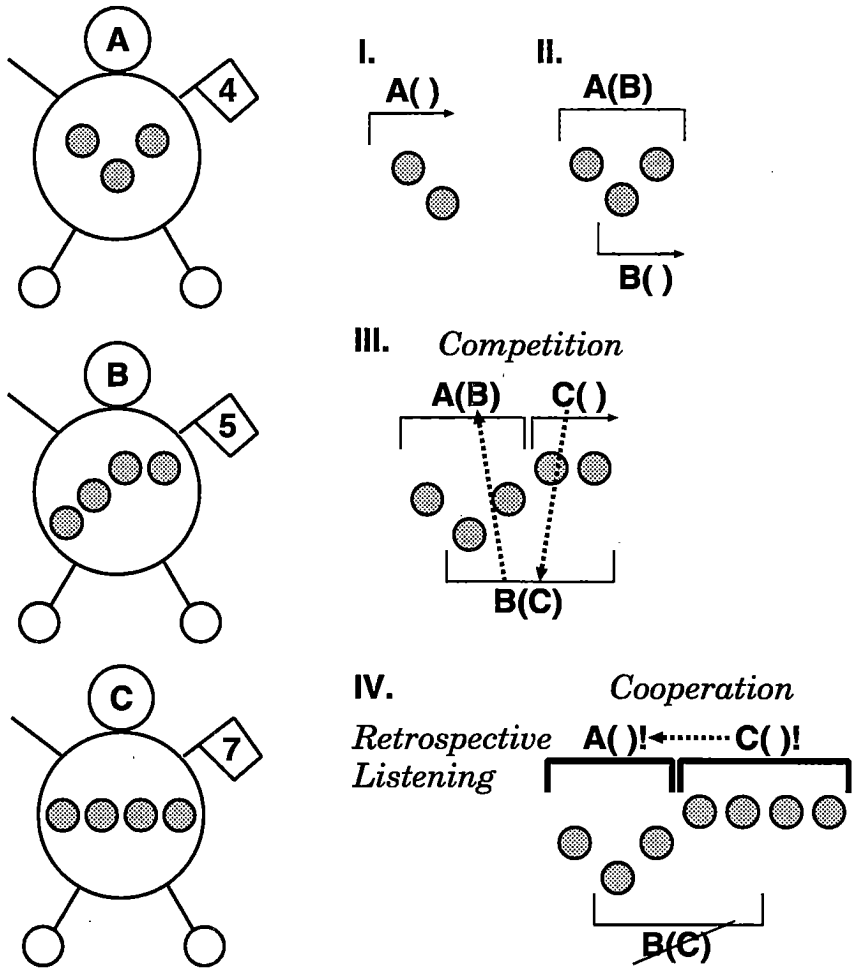


Figure 6.7: A parsing competition showing competition, cooperation and retrospective listening.

- (III) B notices that it is being threatened by agent C (higher frequency) and adds C to its list. This leads to a three-way competition in which A is waiting for B, while B is in turn waiting for C.
- (IV) C completes its match and declares a win because it has no threatening competitors. Upon seeing this, B gives up, clearing the way for A to go ahead and also declare a win.

Thus, the results of a competition can propagate across agents who are not in direct competition with each other through an agent who is a competitor of both. This is referred to as *masking*, as C is said to mask B for A's benefit. In this way, A and C are indirectly *cooperating*.

Minsky proposes that agents can cooperate through one agent lowering the activation threshold of another agent, thus allowing the latter to perform a task in a case where it otherwise would not have been able to [82]. The masking described here is in certain ways similar to this approach.

In the second example above, Agent A waits for a competitor B to decide, while B in turn waits for the results of yet a third agent C. Through this delay in processing, Maestro displays retrospective listening, making its parsing decisions on the basis of information that arrives later in the performance. This is in keeping with Jackendoff's theories of retrospective listening, described in Section 2.2.6.

From the perspective of modelling, this phenomenon can be viewed as an implementation of the echoic memory mentioned in Section 4.1.1.1. Maestro can go back and re-process recent musical events it has heard. No explicit limit is placed on how far back Maestro can go. Instead, the maximal distance allowed is back to the most recent cut-point.

If all threats are removed in one of the above two ways, A's first list becomes empty and it can proceed to win the competition. However, if at any point during A's parsing phase, one of the agents threatening A declares a win, A immediately gives up and terminates. When terminating, it notifies all the agents on its second list that it no longer poses a threat to them.

Only the agent that finds itself with no competitors remaining proceeds to determine the final parse of the system for that portion of the input. Out of the various inter-agent interactions, a desired system-wide parsing behaviour emerges. Examples of Maestro's actual parsing output are shown in Figures 6.8 and 6.9.

The listening agents, with their templates and their internal pointers used to keep track of what portion of the template has matched the input,

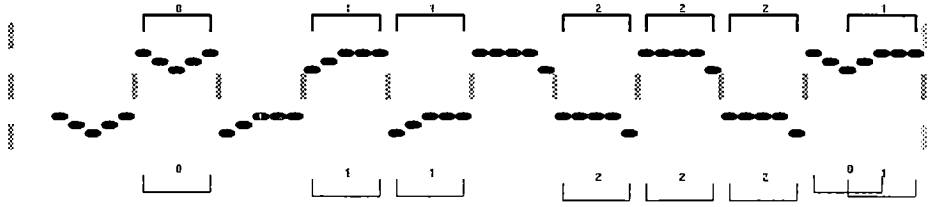


Figure 6.8: An example of a parsing competition. Various parsing attempts are shown below the music, while the successful parses are shown above. Agent 1 has a higher frequency of occurrence than agent 0. Therefore, Agent 1 wins the parsing competition at the end of this selection.

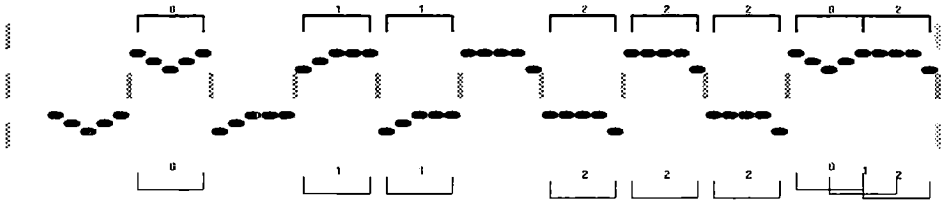


Figure 6.9: Another example of a parsing competition. Agent 2 defeats agent 1, thus clearing the way for agent 0 to be chosen. This shows indirect agent cooperation (between 2 and 0) and retrospective listening (0's claim is delayed).

are in many ways similar to edges in a chart parser described in Section 6.1.2.3. Agents in the matching phase are similar to active edges, while those in the parsing phase are similar to inactive edges. The parsing competition is, in effect, a distributed implementation of breadth first, bottom-up, left-to-right, partial, optimal chart parsing. The primary difference between standard chart parsing and Maestro's parsing methodology is the distributed nature of Maestro's approach. Instead of monolithic, centrally-controlled processing, the individual agents act autonomously, and all decisions are made independently by the individual agents.

The distributed nature of this approach can be argued to be more functionally similar to parallel distributed processing. This is in also line with Jackendoff's [61] suggestion that, all analyses are undertaken, and they are abandoned independently of each other, leaving the field to whatever analyses remain active.

6.2.3 Agents and Cognitive Load

As Maestro's distributed parsing approach relies on the instantiation of multiple listening agents, it is important to address the question of cognitive load: what is the implication of many agents being active at one time? This issue is now addressed from the perspective of musical tension.

When describing his agent-based rhythmic parsing system, Rosenthal [98] states that the number of recogniser agents (Section 11.4.2) waiting to be parsed by the system is correlated to "tension" in music. This issue is also addressed by Jackendoff, who notes that musical complexity can be measured by the degree of embedding and ambiguity, and draws a connection between simultaneous conflicting hypotheses and musical tension [61].

Therefore, there are grounds to view multiple competing agents as correlates of musical tension. The next question to address is then: how many hypotheses can the system maintain simultaneously, without overloading or breaking the constraints of cognitive realism? Jackendoff states that since little is known about space limitations in the brain, it is hard to evaluate how many parallel analyses can be maintained at once. However, Jackendoff states that this limit should be susceptible to experimental study [61].

As no clear limits are available, Maestro does not place any limits on the numbers of agents active in the system at one time. However, Jackendoff notes that the complexity of the Bach analysis presented in [61] puts a surprisingly large lower bound on the number of hypotheses the parser must be able to entertain at once without appreciable stress.

This general view of tension is adopted in this research. Additionally, experiments studying the relationship of agent activation levels to ambiguity are reported in Chapter 8. It is shown that, with training, multiple agents

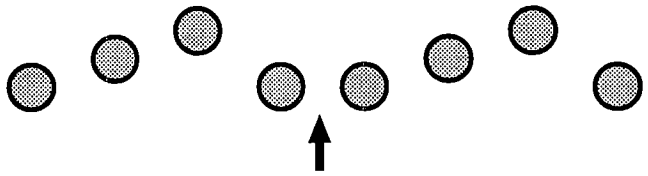


Figure 6.10: The tune *Frere Jaques* shows the necessity for repetition-based segmentation in certain cases where discontinuity-based segmentation is insufficient. The notes shown are C D E C C D E C.

activated together cause "ambivalence" in the system, leading to less risky predictions being made.

6.2.4 The Circularity Problem Revisited

One final issue involved in parsing is the *circularity problem*, introduced in Section 3.2.6 during the discussion of segmentation. There are two methods commonly used for segmenting music: local discontinuities and repetition. While searching for discontinuities can yield very impressive results, Hiraga points out that discontinuity alone is not enough [56]. Both he and Cambouropoulos [20] note that the first two phrases in the popular tune *Frere Jaques* would not be noticed as distinct based on discontinuity alone, but are only noticed by searching for the repeating pattern C D E C (Figure 6.10). Hiraga concludes that identifying repetition is a prevalent and autonomous cognitive process, to be regarded as a primitive operation working at the same level as the segment detectors.

Herein lies the circularity problem mentioned by both Hiraga [56] and Larson [70]: segmentation relies on finding repeating patterns, which in turn relies on segmentation to determine which patterns to look for.

One solution to this would be to keep track of *all* possible patterns up to a certain length, thus not requiring a specific segmentation strategy. Apart from placing a limit on the length of the patterns, this method involves a brute-force full search, and is inefficient. In designing a real-time interactive music system, Rowe notes that such an exhaustive search would quickly overwhelm the available processing [101]. Cambouropoulos's [19] work describes such a full-search approach. However, he readily states that this is not designed to comply with the restrictions of cognitive realism.

Instead of a full-search, Maestro deals with this inherent circularity by performing a *directed search* for patterns. The otherwise overwhelming search space is cut down by examining only those patterns that are suggested by perceptual discontinuity cues – Maestro's segmentation and modelling

stages. This approach thereby integrates both discontinuity-based segmentation and repetition-based segmentation. A similar method of integrating these two is also suggested by Cambouropoulos in [19, pp. 128-9].

According to this approach, patterns can never be found based on repetition alone if no segmentation cues lead to their identification at any point during the piece. However, after the pattern is noticed once due to segmentation cues, it can be noticed later on in the piece even in the absence of segmentation cues during the repetition of the pattern, as Maestro's parsing stage is thereafter able to notice a repetition of the pattern and to label the data accordingly.

When discussing the problem of finding word boundaries in spoken language (related to parsing segmentation ambiguity), Krumhansl and Jusczyk highlight the special ability of a native speaker of a language to analyse utterances into a hierarchy of discrete units despite the fact that pauses between successive units are often absent from the acoustic signal [69, p. 70]. This ability is based on using previous experience and higher level structures to resolve the ambiguity. In a similar way, Maestro's parsing stage uses structures learned from previous experience (i.e. the patterns stored in its model) to resolve parsing segmentation ambiguity in music.

6.3 Summary

This chapter has reviewed Maestro's parsing capabilities. Parsing is the process through which people are said to achieve understanding of a piece of music. It involves organising the musical surface according to the listener's internal model. Musical parsing is in many ways related to natural language parsing. Of the various types of ambiguity present in natural language parsing, parsing segmentation ambiguity, related to repetition-based segmentation, is shown to be relevant to Maestro's handling of music. Maestro handles this ambiguity in its parsing stage.

People parse deterministically; when dealing with parsing ambiguity on-line, they must sometimes resolve an ambiguity before sufficient information becomes available to do so conclusively. Of the selection factors commonly used by people in resolving natural language ambiguity, right association and lexical preference are shown to be related to Maestro's handling of music. Maestro employs both of these criteria in resolving ambiguity. Jackendoff proposes the parallel multiple analysis model for musical parsing, capable of maintaining multiple simultaneous hypotheses on-line when faced with ambiguity. Maestro implements certain key aspects of Jackendoff's theory.

Maestro's parsing is performed in a distributed fashion by the various listening agents. It is, in effect, a distributed implementation of breadth first, bottom-up, left-to-right, partial, optimal chart parsing. Various agents

compete and cooperate with each other, while each pursues its own parsing goal. Out of the interactions between the individual agents, the desired system-wide parsing behaviour emerges.

Maestro addresses the circularity problem raised by Hiraga and Larson by performing a directed search for patterns, based on its perceptually guided segmentation.

Chapter 7

Single-Style Results

With all four stages of Maestro's design fully described, the large scale experiments in music learning for which Maestro was primarily designed are now presented. This chapter presents experiments involving music from a single style. As a means of continuing from where previous research left off, experiments are performed with the same 100 Bach Chorale melodies used by Conklin and Witten. Results are compared with human data and performance is studied both on the level of single notes and on the level of entire pieces. Learning experiments with much larger data sets are also performed and multiple-step-ahead predictions are studied. A three-fold framework for analysing music learning data is developed and tested, measuring context model growth, the number of predictions generated and the prediction performance.

7.1 Bach Chorales

Conklin and Witten [132] analysed the final state prediction capabilities of a context-model-based music learning system after it had listened to nearly 100 Bach Chorales. Their results show that their machine model performs almost as well as human listeners, and that the responses of the machine model and of humans to various musical selections are highly correlated. A first set of experiments was performed here with the same data to continue from where Conklin and Witten left off.

7.1.1 Comparison With Previous Work

In [132], Conklin and Witten compare the prediction performance of their machine model to that of humans. A data set containing 100 Bach Chorales (selected from Bach Chorales `bc1` through `bc285`) is used. Out of the data

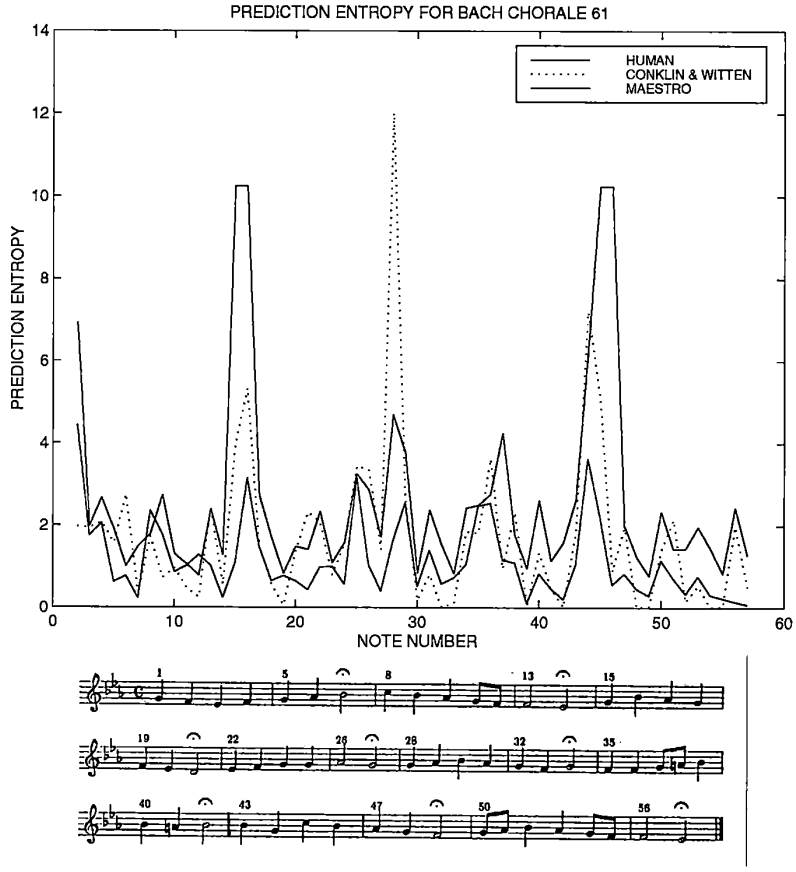


Figure 7.1: Note-by-note prediction entropy for Bach Chorale `bc61`. Three sets of results are compared: the human and machine data reported by Conklin and Witten [132], and Maestro's data.

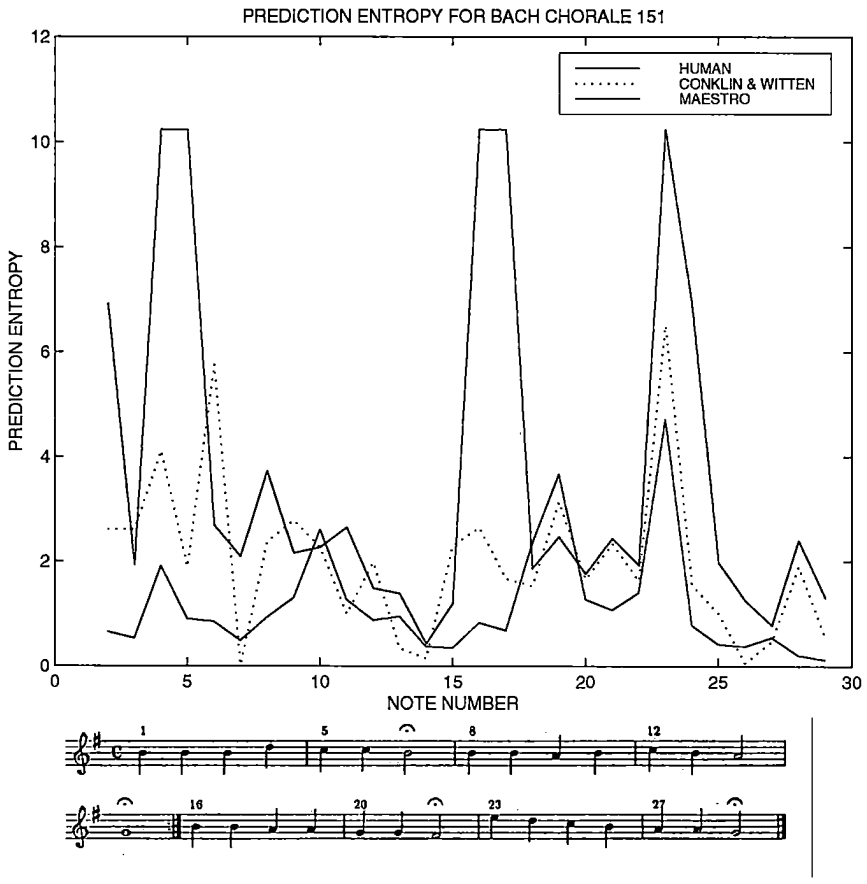


Figure 7.2: Note-by-note prediction entropy for Bach Chorale `bc151`. Three sets of results are compared: the human and machine data reported by Conklin and Witten [132], and Maestro.

	Average Prediction Entropy	
	Chorale `bc61`	Chorale `bc151`
Human	1.1357	1.1622
Conklin and Witten	1.7362	2.0326
Maestro	2.6627	3.7596

Table 7.1: Average prediction entropy for Bach Chorales `bc61` and `bc151`. Results are compared for the human and machine data reported by Conklin and Witten [132], and for Maestro.

set they choose two (Chorales `bc61` and `bc151`) to serve as the test set. After training their machine on all the other Chorales, they calculate the note-by-note *machine* prediction entropy for Chorales `bc61` and `bc151`. Prediction entropy is used as a measure of prediction performance, as introduced and described in depth above in Chapter 5. Recall that a high prediction entropy reflects poor prediction performance, while a low prediction entropy indicates good prediction performance. Conklin and Witten compare these results with experiments in which human subjects were asked to perform the same prediction task, and a corresponding set of *human* prediction entropy values were calculated. (The reported human data is actually a weighted average of a number of listeners, as reported in [77] and [132].)

Conklin and Witten's results show that the average human level of performance is better than that of the machine. This is to be expected, as their system deals with only 100 Chorales, while the human subjects have had decades of prior listening experience. Additionally, the machine is only a limited, computational model, which can by no means be compared with the complex, powerful and little-understood capabilities of a human brain. However, the important outcome discussed by Conklin and Witten is that the two sets of results are correlated: the machine model does well where humans do well, and does poorly where humans do poorly [132].

In the first experiments reported here, in order to compare Maestro's performance with these results, Maestro was presented with the same training and test Chorales used by Conklin and Witten, and the prediction entropy was calculated. Before presentation to Maestro, the extra information about key signature, time signature and fermatas was removed (See Appendix C). While Conklin and Witten make use of this information to aid in prediction, its inclusion here would violate Maestro's realistic input specification, as discussed in Section 2.2.1.

Figures 7.1 and 7.2 show the results compared for the human and machine data reported by Conklin and Witten (kindly provided by Darrell Conklin), and for Maestro.

First, the relative levels of performance are examined. Table 7.1 shows

a comparison of the average levels of prediction performance. In both cases Maestro performs less well than both the human results and Conklin and Witten's machine models. This is to be expected. As described in Section 3.1.1, Conklin and Witten build their model storing every possible context segment of up to length three. Additionally, they include *a priori* stylistic information such as tonality and rhythm to improve their machine's performance, as described later in Section 11.2.4. In contrast, Maestro builds its model in a less brute-force way, according to more realistic cognitive constraints, and also uses no *a priori* stylistic information in making its predictions. Thus, memory usage is much lower in Maestro, where only specific PGS-derived segments are stored and analysed as opposed to all the possible segments as in Conklin and Witten's work. Also, Conklin and Witten's system chooses between 20 possible notes [77] that appear in the Chorale melodies, while Maestro chooses between 121 possibilities in order to maintain greater flexibility, thus starting at a much higher chance prediction entropy level (6.92 for Maestro vs. 4.32 for Conklin and Witten). Finally, Conklin and Witten integrate predictions made from a short term context model with those generated from long term context model. While this approach would improve Maestro's predictions as well, long term predictions in Maestro are studied alone in order to better focus on the long-term learning occurring in the system.

It is therefore to be expected that Conklin and Witten's optimised machine prediction model results in better prediction performance than Maestro, the design of which is more focused on the flexibility and the constraints of cognitive realism. The second question to explore then is to what extent Maestro's predictions are correlated with the human results.

The correlation coefficients between the various sets of results are calculated. For the case described here, where two sets of prediction entropy results are represented by two one-dimensional vectors $\mathbf{x}$ and $\mathbf{y}$, the correlation coefficient r_{xy} is defined as:

$$r_{xy} = \frac{\sigma_{xy}^2}{\sigma_x \sigma_y} \tag{7.1}$$

where the covariance σ_{xy}^2 is given by:

$$\sigma_{xy}^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})(y_i - \overline{y}) \tag{7.2}$$

and where N is the length of the vector (i.e. number of notes in the Chorale,

Correlation Coefficients		
	Chorale `bc61`	Chorale `bc151`
Human vs. Conklin and Witten	0.5280	0.5979
Human vs. Maestro	0.5233	0.2797
Conklin and Witten vs. Maestro	0.4754	0.5405

Table 7.2: Correlation coefficients calculated from the data in Figures 7.1 and 7.2, comparing the human and machine data reported by Conklin and Witten [132], and Maestro.

the same for both vectors) and $\bar{x}$ is the average for the vector (average prediction entropy for that set of results). The standard deviation σ_x is given by:

$$\sigma_x = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{7.3}$$

Table 7.2 shows the correlation coefficients calculated for the various prediction results. For Chorale `bc61`, Maestro shows about the same high level of correlation with the human data as does Conklin and Witten's model (0.5233 and 0.5280 respectively). It is impressive that Maestro, despite its constraints, is able to achieve the same general level of correlation with the human data. It is proposed that although Maestro is slower in building up its model and does not incorporate any style specific knowledge, its more cognitively realistic modelling strategy causes its prediction performance to be similar to that of humans.

For Chorale `bc151`, however, Maestro shows a lower (but still positive) correlation with human results than does Conklin and Witten's model (0.2797 and 0.5979 respectively). This result is to be expected due to the limitations of Maestro's model stated above.

The variability in correlation results between the two Chorales highlights the fact that a test set containing two Chorales is indeed very little data for determining correlations between different sets of prediction results. Data from many more Chorales would be needed in order to calculate a set of conclusive figures. However, Manzara *et al.*'s original experimental work [77], which was the basis for Conklin and Witten's experiments, produced measurements for only two Chorales. Human data of this sort is extremely laborious to obtain, requiring many subjects and long hours of experimentation. Still, it is fortunate that these human results are available at all, keeping in mind that, as discussed in Chapter 5, the only absolute benchmark for evaluating prediction performance levels is direct comparison with human results.

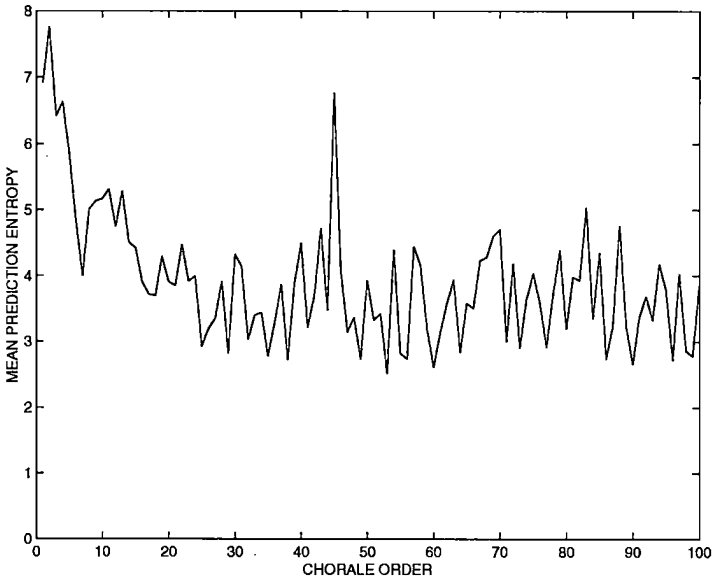


Figure 7.3: Average prediction entropy per Chorale as Maestro listens to 100 Bach Chorales in series. The Chorales were presented to Maestro in a certain order (`MIX 1`), and the numbers 1-100 on the X axis simply correspond to the order of presentation (rather than the actual names of the specific Chorales).

This first set of results helps to place Maestro in its proper research context by comparing its performance with experiments conducted by previous researchers. Additionally, the direct comparison with human data helps to validate Maestro somewhat as a functional model of certain aspect of music cognition and learning, capable of producing meaningful results. Further experiments are described below that would be difficult to perform with human subjects, and so validation would be even more difficult. The results of these later experiments therefore can be considered in light of this initial validation study.

7.1.2 Learning Process

Conklin and Witten study the *final-state* prediction capabilities of a context-model-based music learning system, after it had listened to nearly 100 Bach Chorales. It is also interesting, however, to study the learning process itself. In what ways does the system's performance change as it gains experience? As Maestro is designed to be a more cognitively realistic model of certain aspects of music learning, it can be used to investigate these issues experimentally. The results obtained in these experiments are interpreted in light of the theories embodied in Maestro's design.

In this set of experiments, the 100 Bach Chorales used by Conklin and Witten were given to a fresh instantiation of Maestro, and the performance was measured using prediction entropy. Figure 7.3 shows the average prediction entropy per Chorale, over the course of listening to 100 Chorales in

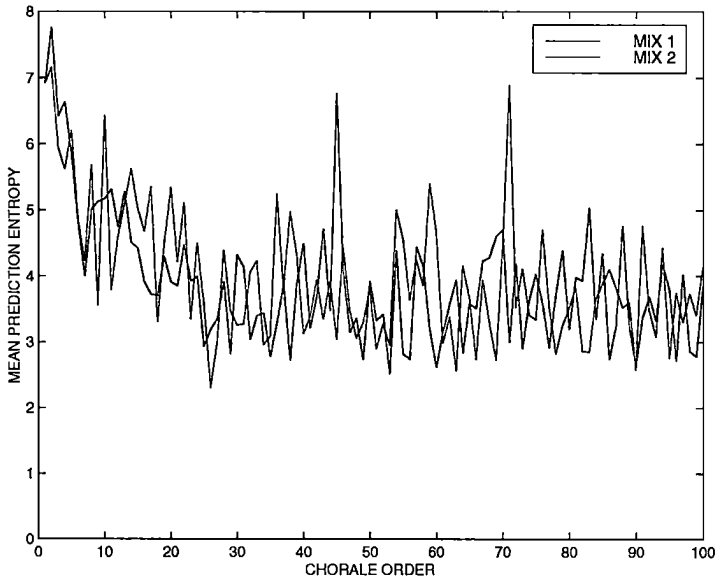


Figure 7.4: The same prediction entropy results after randomising the order of the Chorales (`MIX 2`), compared with results of Figure 7.3 (`MIX 1`).

series. The Chorales were presented in a certain order, called here `MIX 1`. Note that unlike the results reported above on note-by-note basis, these and the remaining results reported in this chapter are reported on an average, per-Chorale basis.

A gradual decrease in prediction entropy can be noticed in the beginning of the graph, indicating that the system is learning with experience. However, due to the large inter-song variability in prediction entropy, studying the actual learning process is difficult.

It is postulated that this variability arises from the intrinsic entropy of the specific Chorales (see Section 5.2.4). To verify this, the 100 Chorales were shuffled around (`MIX 2`) using a pseudo-randomiser program. The experiment was then performed again.

Figure 7.4 shows that while the general learning trend remains, inter-song variability is indeed specific to the Chorale: the largest peak in both cases represents the average prediction entropy resulting from Chorale reference number `bc120`. Despite its different place in the order of presentation, the inherent entropy of `bc120` causes the prediction of the notes to be more difficult, resulting in higher prediction entropy. This confirms that the inter song variability is in large part a result of the inherent entropy of the individual songs.

Before proceeding with further experiments, the issue of data visuali-

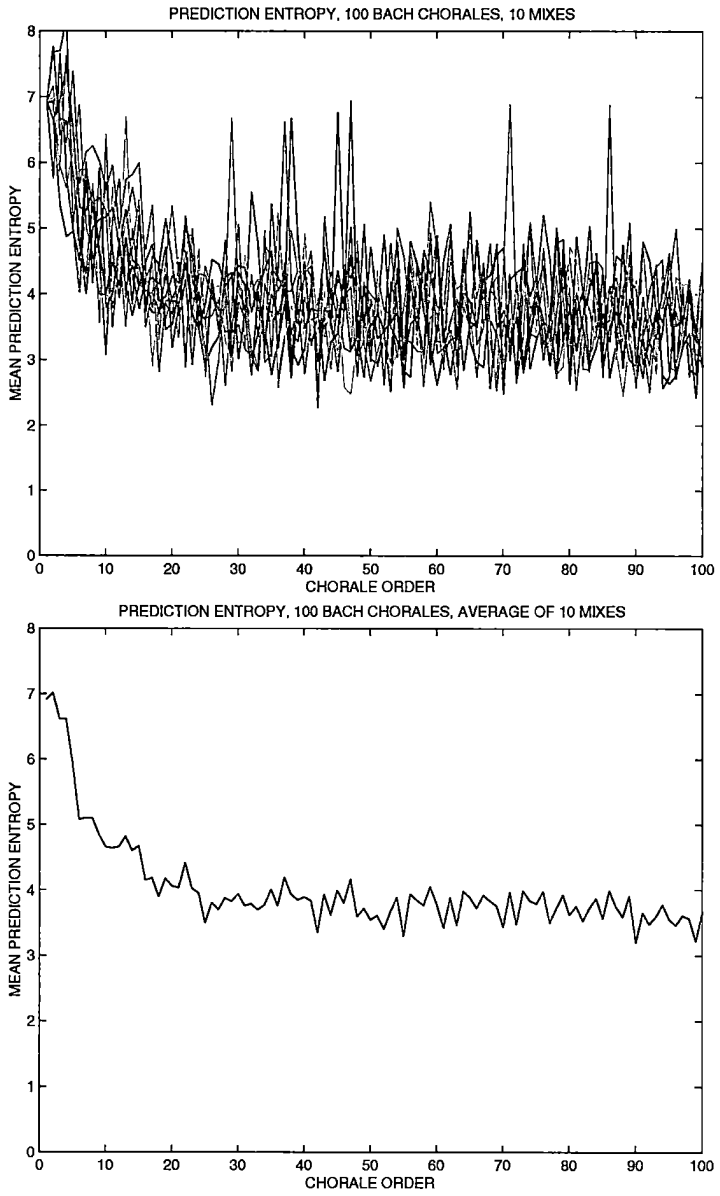


Figure 7.5: Ten re-shuffled runs of 100 Bach Chorales (top), and the resulting average (bottom).

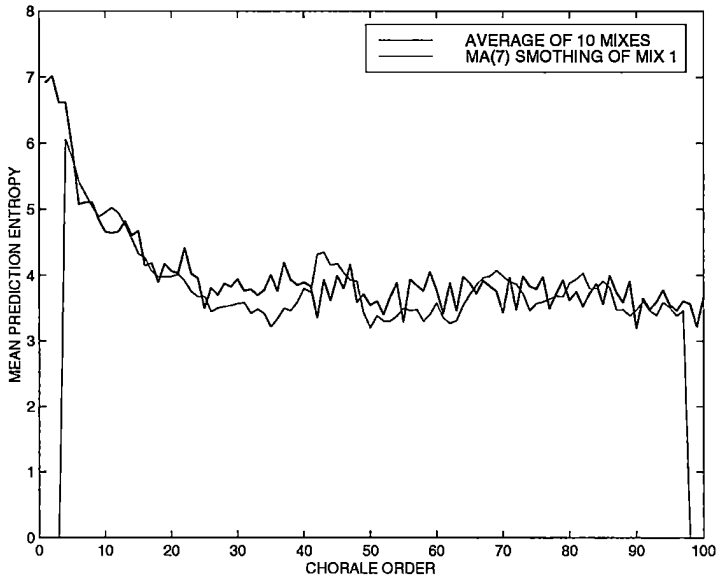


Figure 7.6: Moving-average smoothing of one Bach chorale, compared with the average of ten runs from Figure 7.5. `MA(7)` indicates that moving-average smoothing was performed with a sliding window size of seven.

sation needs to be addressed: given the inter-song variability, what is the best way to view the general learning trend? One approach is to average together many runs, each based on a different ordering of Chorales. Figure 7.5 shows ten such runs overlaid on each other (top), and then averaged together (bottom). The averaging almost completely removes the inter-song variability, and a smoother learning curve remains.

Another method of visualising the learning trend is also explored: smoothing by moving-average. Conklin and Witten use smoothing for visualising medium term prediction entropy trends within pieces [132, p. 78]. Specifically, they use a seven-note, triangular sliding window for smoothing. Here, a flat window of appropriate length is used, as specified in the various graphs below.

Figure 7.6 shows that moving-average smoothing yields similar results to averaging many runs. However, moving-average smoothing leads to loss of samples at either end of the data. The wider the smoothing window, the better the fit to that of averaging many runs, but the more samples are lost towards the edges (Figure 7.7). Moving-average smoothing is used for the remaining experiments.

The experiments performed with the Bach Chorales allow the present research to pick up where Conklin and Witten left off. Extending Conklin and Witten's work that analysed only the final state prediction performance

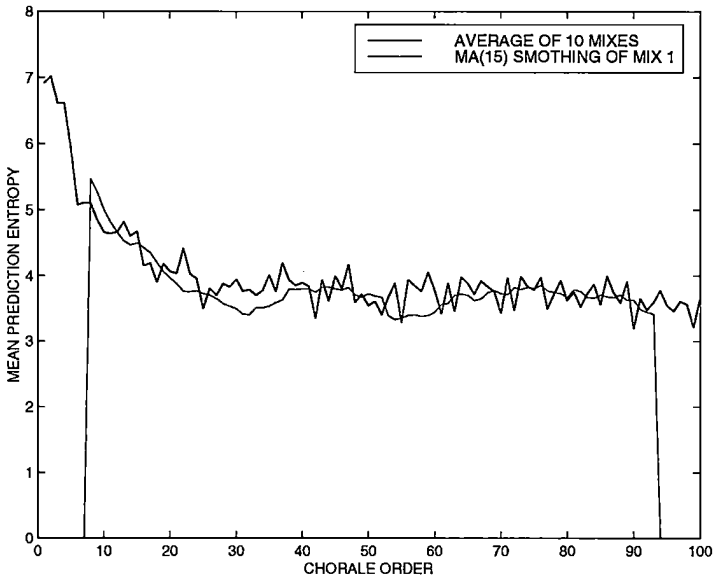


Figure 7.7: A wider smoothing window leads to a better fit, but more samples are lost.

of the system, the experiments here reveal a clear learning process occurring during training. To the author's best knowledge, this type of analysis has not been reported elsewhere, and thus constitutes a major contribution of this research.

In the graphs presented so far, it appears that learning drastically slows down after the first 30 Chorales. The experiments reported in the next section show that significant learning continues to take place long after these first few songs. This confirms that in order to study the full learning process, larger data sets must be used.

7.2 Essen Folksong Collection

To meet the need for larger data sets, the *Essen Folk Song Collection* (EFSC) [106] was used. The collection is the result of the life work of the late Professor Helmut Schaffrath at Essen University in Germany. It is currently distributed by the Centre for Computer Assisted Research in the Humanities (CCARH) at Stanford University [107]. The EFSC contains over 5,100 German folk songs, and over 600 other folk songs from around the world. An additional 2,200 Chinese folk songs, not yet in the standard distribution, were also obtained with the kind help of Don Anthony and Eleanor Selfridge-Field at CCARH.

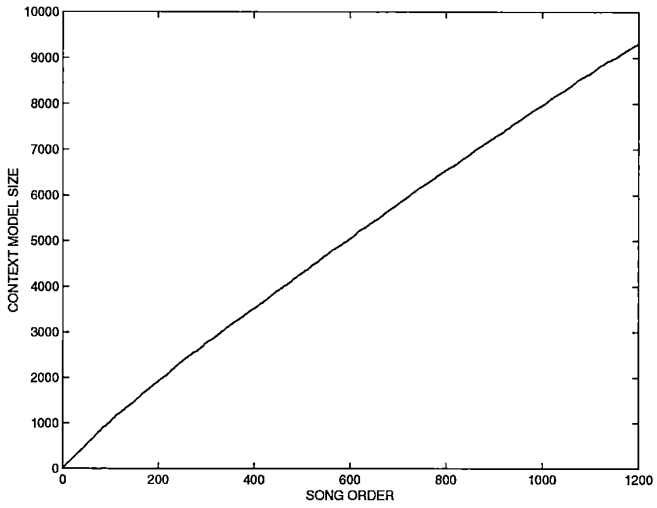


Figure 7.8: The size of the LTM context model as Maestro listens to 1,200 German folk songs.

The large number of tunes from different styles allow large-scale stylistic learning experiments to be performed. These corpora constitute a more realistic musical experience base, enabling a fuller study of the music learning process. Additionally, longer range trends are less affected by intra-style (inter-tune) variability, making visualisation easier.

To begin, a collection of 1,200 German folk songs was used (EFSC reference numbers `deut0567-deut1766`). These were presented to a fresh instantiation of Maestro, and results were collected throughout the system's processing of the 1,200 songs. In order to conduct a thorough study of long-term music learning, a framework for studying music learning was developed over the course of this research. Three different types of analysis were performed and are now presented in turn: context model growth, the number of predictions made, and the prediction performance.

7.2.1 Context Model Growth

Figure 7.8 shows the overall size of Maestro's LTM context model over the course of listening to the 1,200 folk songs. Maestro learns by adding segments to its context model, and the increase of model-size with experience is readily evident from the graph.

Maestro's context model consists of segments of various lengths. Graphs of model size by segment length are shown in Figure 7.9. Different model sizes are evident for the different segment lengths. The number of six-interval-long segments grows the fastest and is the largest.

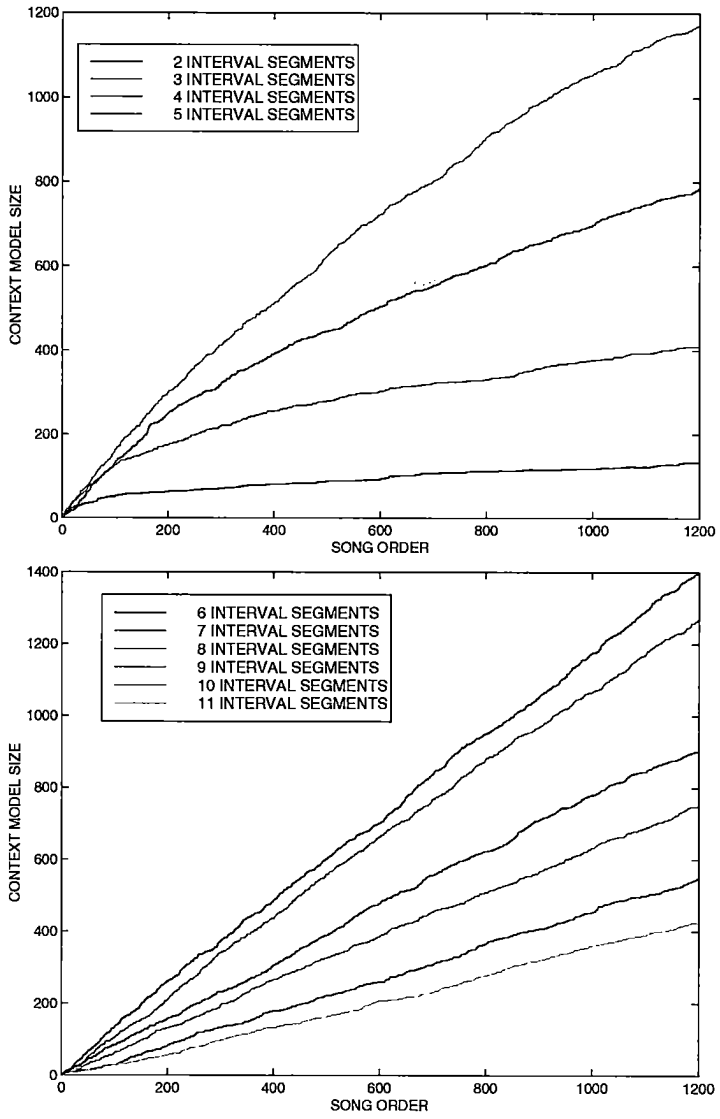


Figure 7.9: The size of the LTM context model according to different segment lengths, as Maestro listens to 1,200 German folk songs.

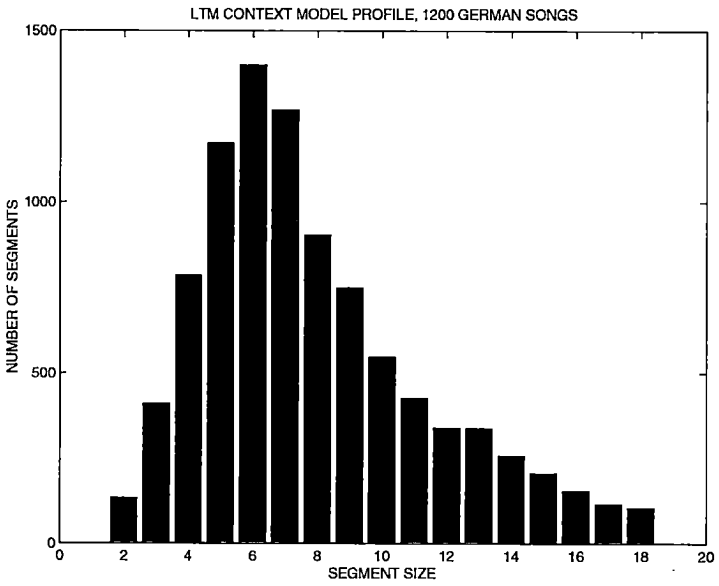


Figure 7.10: The size of the LTM context model as Maestro listens to 1,200 German folk songs.

To provide another perspective, the final state histogram of model size by segment length is shown in Figure 7.10. The number of segments of each length depends on Maestro's segmentation stage, which generates more candidate segments of some lengths than of others. The histogram of context model size is analysed in more depth in Section 9.4.

The development of the context model can be examined further by looking at the context model *growth rate* – the rate of addition of new segments to the model. As the number of notes per tune varies between the various folk songs, calculating the model growth rate per song would not be appropriate. Instead, the context model growth rate is calculated as the average number of new segments added per note-event processed.

Figure 7.11 shows the context model growth rate per note-event processed for the 1,200 folk songs, averaged for each song. The graph clearly shows that with increased training on a homogeneous corpus, fewer new segments are added to the model: the model captures many of the stylistically common segments present in the earlier songs. When it sees them again later, it does not need to add them as new segments, but instead simply increments the frequency counts of the segments already stored in the context model. Therefore, this learning process is reflected by a decreasing context model growth rate.

Figure 7.12 shows the context model growth rate by segment length. The number of shorter segments is saturated sooner (indicated by decreas-

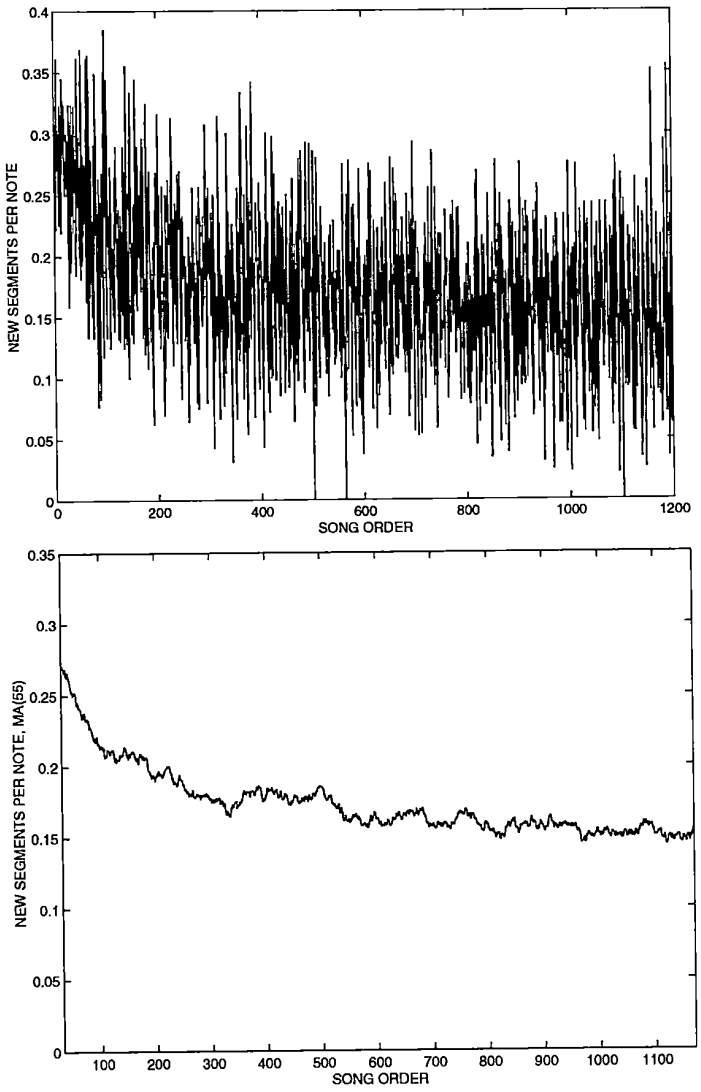


Figure 7.11: The rate of context model growth as Maestro listens to 1,200 German folk songs (top), also shown smoothed to better reveal the trend (bottom).

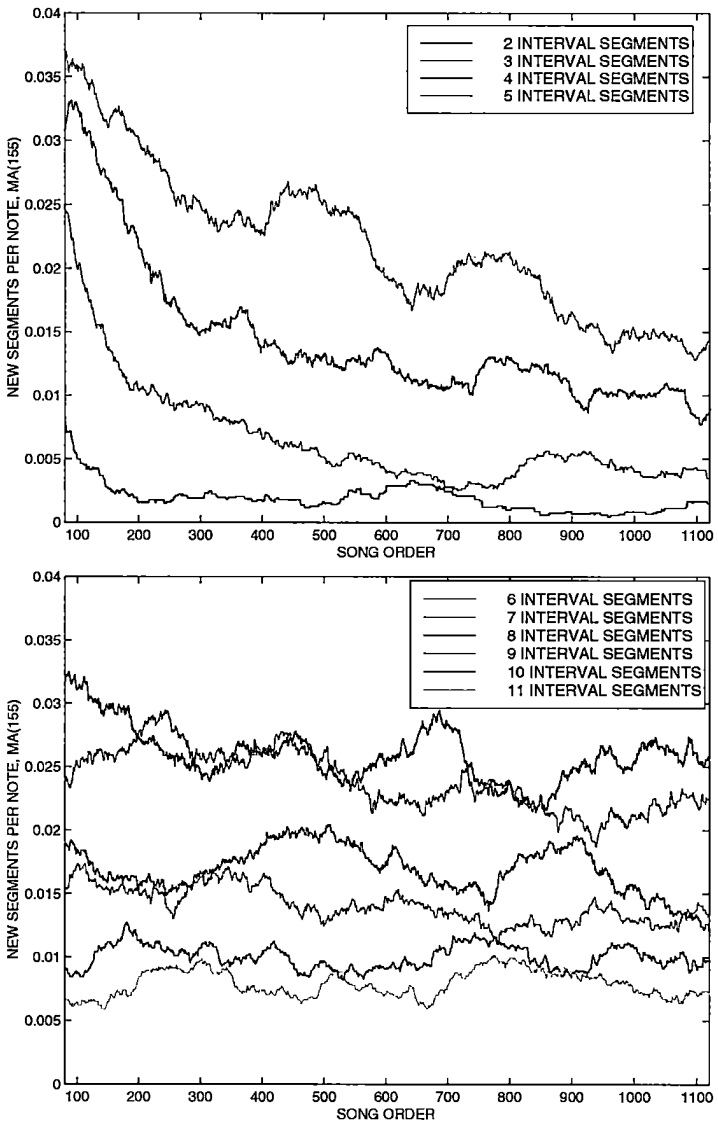


Figure 7.12: The rate of context model growth, according to different segment lengths.

ing growth rate), as there are fewer possibilities for shorter segments (the combinatorial space of possibilities is smaller). In contrast, the number of longer segments continues to grow at a fairly steady rate.

7.2.2 Number of Predictions Made

Maestro generates predictions from the segments stored in its context model. As its context model grows with training, Maestro is able to generate more predictions in a larger number of musical contexts.

The average number of one-step-ahead predictions made per note-event is shown in Figure 7.13. As expected, with increased training, more predictions are made. This effect is visible during the first 50 songs or so, and the early saturation of this phenomenon is addressed further in Section 7.2.4 below. However, even after Maestro is well trained, there are still certain difficult points in some songs, during which no predictions are made.

7.2.3 Prediction Performance

In order to gauge the quality of Maestro's predictions, the prediction entropy is calculated. Figure 7.14 shows Maestro's one-step-ahead prediction entropy when listening to the 1,200 German folk songs. The dashed line at 6.92 indicates chance level – when the prediction entropy is based on a flat probability distribution. This occurs when no information is available from the contexts in the model in order to make a prediction.

Though starting at the level near chance, prediction performance then improves with training, exhibiting an exponential-like learning curve. A more complete picture of the learning process can be seen in these results, extending well beyond the first 100 songs.

It is also significant to note how the gradual reduction in one-step-ahead prediction entropy is qualitatively similar to the gradual decrease in context model growth rate seen in Figure 7.11. As the system gains familiarity with the style, fewer new segments are added and better predictions are made. These two effects both exhibit quick improvement initially, and with time gradually settle to a steady state.

7.2.4 Multiple-step Predictions

As described in the preceding chapters, due to the variable-length contexts stored in its context model, Maestro is capable of generating appropriate multiple-step-ahead predictions. The number of predictions generated and the prediction entropy are now both re-examined for multiple forecast horizons.

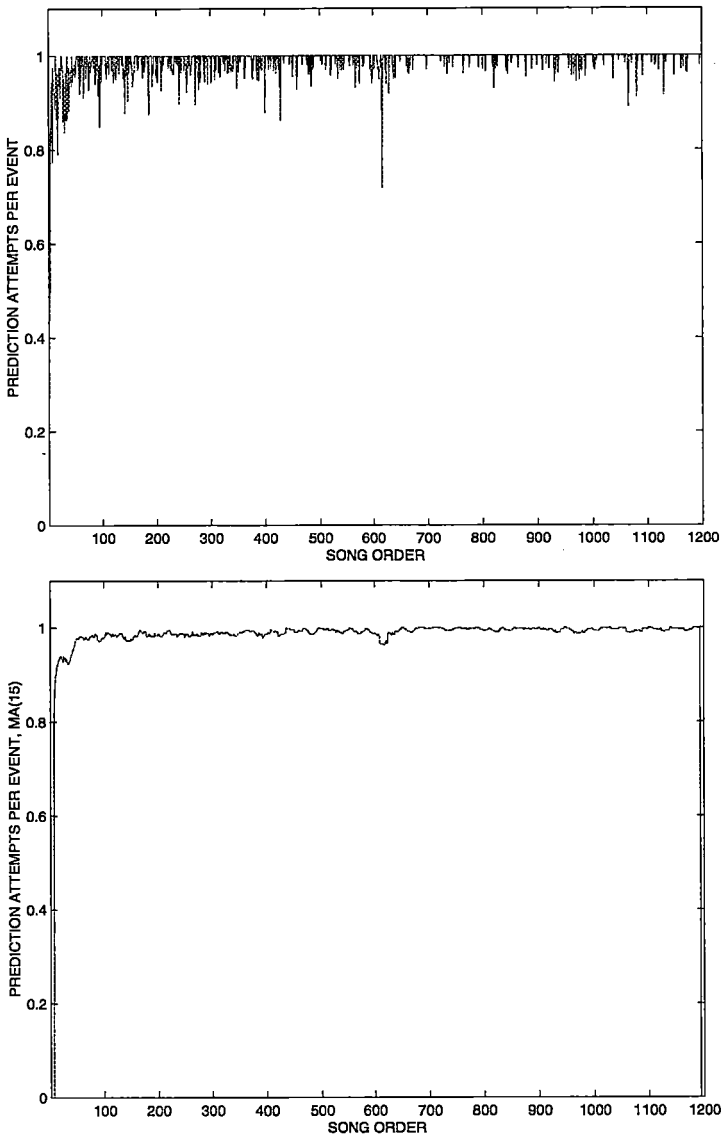


Figure 7.13: The proportion of note-events on which one-step-ahead predictions are made, as Maestro listens to 1,200 German folk songs (top), also shown smoothed to better reveal the trend (bottom).

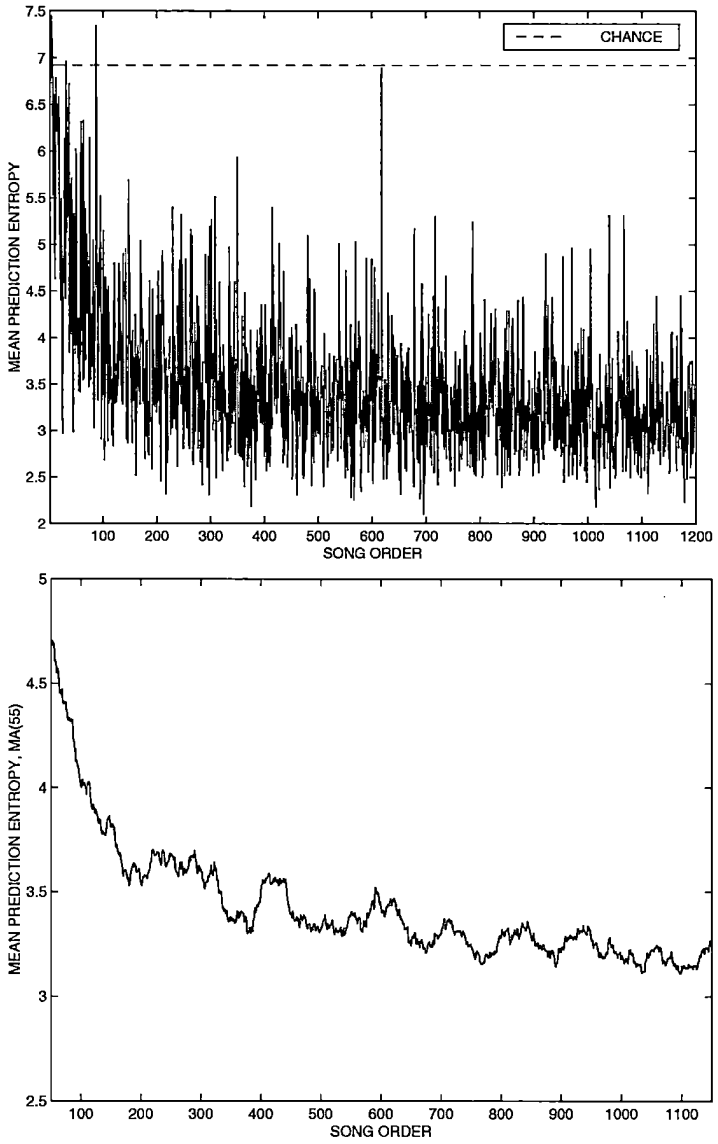


Figure 7.14: The average prediction entropy per song for one-step-ahead predictions, as Maestro listens to 1,200 German folk songs (top), also shown smoothed to better reveal the trend (bottom).

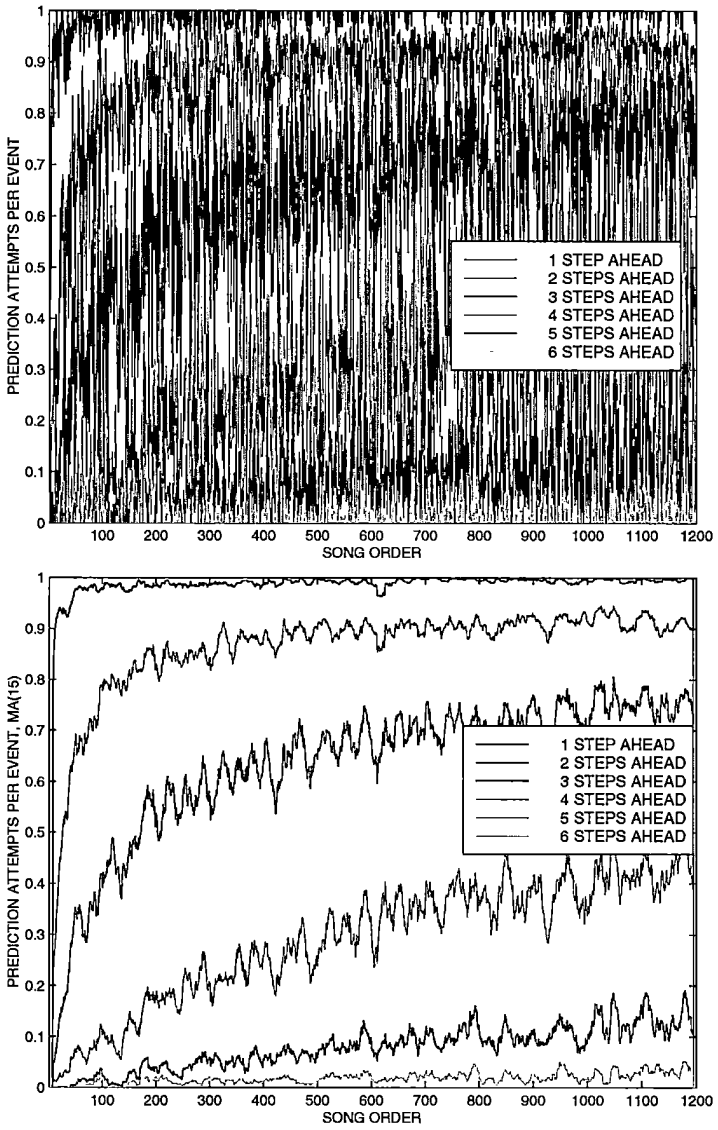


Figure 7.15: The number of predictions made for different horizons as Maestro listens to 1,200 German folk songs (top), also shown smoothed to better reveal the trends (bottom).

Figure 7.15 shows the number of predictions made for various forecast horizons. Within each forecast horizon, the number of predictions increases with training, showing that the short learning trend seen above for one-step-ahead predictions (Figure 7.13), carries through to longer-range predictions.

An additional trend is also visible. As forecast horizon length increases, the number of predictions decreases and the rate of growth of the number of predictions slows down. Longer-range predictions rely on larger contexts being stored in the model, as well as longer matches between these segments and the input. It is therefore understandable that fewer longer range predictions are generated.

The prediction entropy for various forecast horizons is shown in Figure 7.16. The prediction quality decreases with increasing forecast horizon. This effect is to be expected, as it is generally more difficult to predict what will happen further in the future. Additionally, the quality of predictions increases with training within each forecast horizon.

The prediction entropy for four and five steps-ahead predictions stays near the chance level (6.92). This observation indicates that for the present learning strategy and corpus size, the maximum useful forecast horizon seems to be three steps ahead. This does not imply, however, that contexts longer than three events are of no use. Recall that in order to generate a three-step-ahead prediction, a context of length six or higher is needed. This is because, as mentioned in Chapter 5, a listening agent waits until at least half of its template has been matched before generating a prediction. Many useful one to three-step-ahead predictions can also be generated from the final portions of much longer contexts stored in the model. (These predictions will be assigned very large *certainty* weights, as described in Section 5.3.2.) Therefore, even though only predictions of up to length three are useful in the present set-up, contexts of all lengths are still important.

Three methods of analysis have been used here to analyse the results: the context model growth rate, the number of predictions made by forecast horizon, and the prediction performance by forecast horizon. Together, these serve to characterise the process of music learning. They are proposed as a general framework for the study of music learning, and are used to analyse the results of further experiments in the following chapters.

7.3 Summary

This chapter has presented experiments performed with music from a single style. Extending the work of Conklin and Witten, Maestro was used to study the learning process that occurs when listening to 100 Bach Chorales. A positive correlation between Maestro's results and those obtained from humans helps to validate Maestro as a functional model of certain aspects of

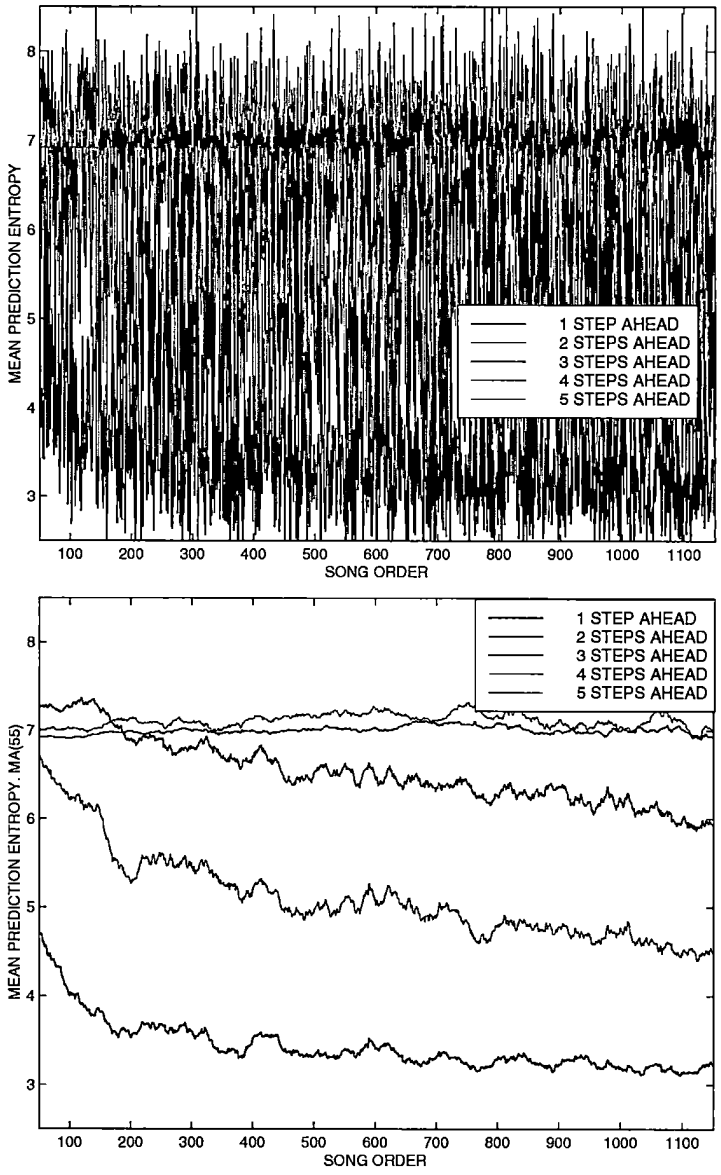


Figure 7.16: The mean prediction entropy per song for different horizons as Maestro listens to 1,200 German folk songs (top), also shown smoothed to better reveal the trends (bottom).

music learning. The music learning process is formally studied, and moving-average smoothing is shown to help in visualising trends in light of inter-song variability.

In order to study a more complete learning process, much larger-scale music learning experiments were conducted with 1,200 German folk songs from the Essen Folk song collection. Multiple-step-ahead predictions are also investigated. Prediction performance improved with training, but decreased with longer forecast horizons.

Three methods are used to analyse the results: context model growth rate, the number of predictions generated, and the prediction performance. Together, these three methods are proposed as a general framework for studying the process of music learning.

Chapter 8

Ambiguity Analysis Results

As described in Section 2.2.5, ambiguity is an essential aspect of music listening, especially in the context of learning. This chapter presents an experimental study of musical ambiguity performed with Maestro.

Three measures of system activity were introduced in Chapters 5 and 6, the first two of which depend on the probability density function (PDF) generated by Maestro's prediction stage:

1. Overall Entropy – related to the sharpness of the prediction PDF.
2. Prediction Entropy – related to the accuracy of the prediction.
3. Agent Activation – the number of agents active at one time.

The relationships between these various measures can be used to investigate certain aspects of Maestro's behaviour, particularly its handling of ambiguity. A collection of 600 German folk songs were used for these experiments. (EFSC reference numbers `deut0567-deut1166`). The songs contain roughly 26,000 individual note-events, and the following plots present results consisting of note-by-note data points (not averaged together by piece).

8.1 Overall Entropy vs. Agent Activation

Maestro was presented with 600 German folk songs and the overall entropy and agent activation were recorded for each note processed. Figure 8.1 shows a scatter plot of the overall entropy plotted against agent activation for each of the notes in the 600 folk songs.

It is clear from the plot that the more agents that are active in the system, the less sharply focused the PDF becomes. This is the expected behaviour, as the different agents make independent, often conflicting predictions, thereby flattening the PDF. What is not so obvious, however, is

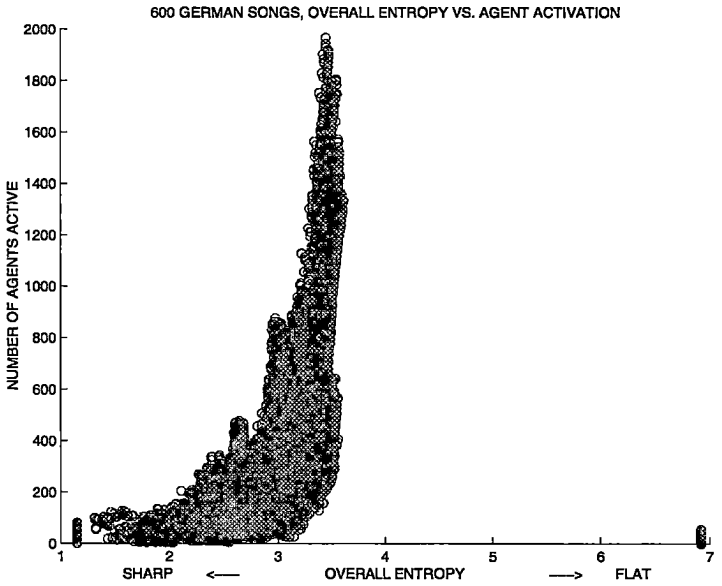


Figure 8.1: A scatter plot of overall entropy against agent activation for 600 German folk songs.

that there is a limit to the flatness. After a certain point, despite the increasing number of agents, the overall entropy asymptotically reaches an upper bound of around 3.5. This is likely due to the fact that since all the agents' patterns have been derived from the same corpus, many of them make similar predictions. Therefore, after a certain point in training, the additional agents make similar predictions to existing ones and the PDF does not get significantly flatter.

The PDF only becomes totally flat (i.e. chance: overall entropy equal to 6.92) when very few agents are active and no predictions are generated. In such a case, the flat distribution (introduced to deal with the zero-frequency problem, Section 5.3.2) ends up being blended with an empty distribution, resulting in yet another flat distribution.

The development of the relationship between overall entropy and agent activation can be studied over time. Figure 8.2 shows four scatter plots of overall entropy vs. agent activation sampled over specific time slices during training. It is evident that with experience, the PDFs become less sharp. As the model becomes better trained, more agents are instantiated, generating more predictions. As the various predictions are often different, this leads to flatter PDFs.

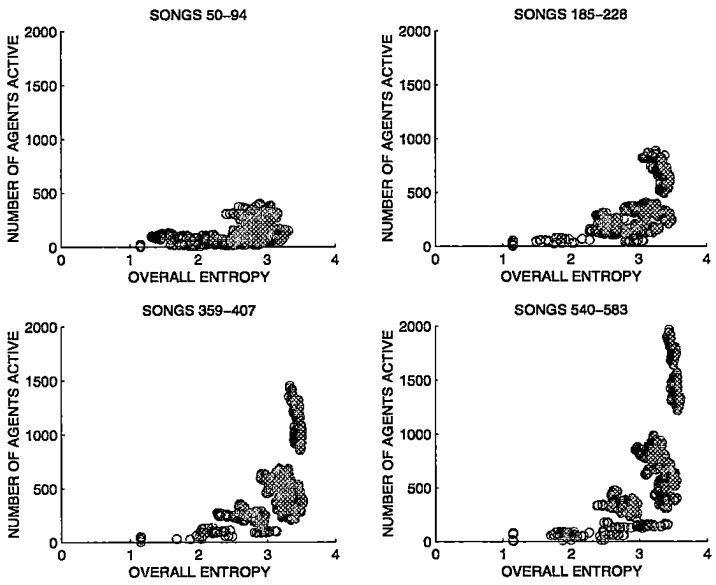


Figure 8.2: The development over time of a scatter plot of overall entropy against agent activation for 600 German folk songs, shown at four progressive stages throughout the experiment.

8.2 Prediction Entropy vs. Agent Activation

Having examined the relationship between overall entropy and agent activation, the focus now moves to the relationship between prediction entropy and agent activation. Figure 8.3 shows a scatter plot of prediction entropy versus agent activation. The collection of points at prediction entropy equal to 10.24 represent those note-events for which none of Maestro's predictions were correct. The data-points at prediction entropy equal to 6.92 represent those note-events for which no predictions were generated by Maestro's context model and a flat distribution was used.

The only trend clearly visible in this scatter plot is that the lower limit of prediction entropy increases with the number of agents (the near-vertical trend seen on the left-most side of the graph). Before analysing this trend, since the scatter-plot does not reveal relative densities of highly crowded regions, it is necessary to display the data from a different perspective.

To permit further analysis, the data in Figure 8.3 was divided into 100 horizontal slices (different ranges of agent activation), and an average prediction entropy was calculated for each slice. The results are shown in Figure 8.4. As before, with increasing agent activation, the prediction entropy gets slightly higher. However, with this view of the data, another trend becomes visible. At low agent activation levels, the prediction entropy becomes dra-

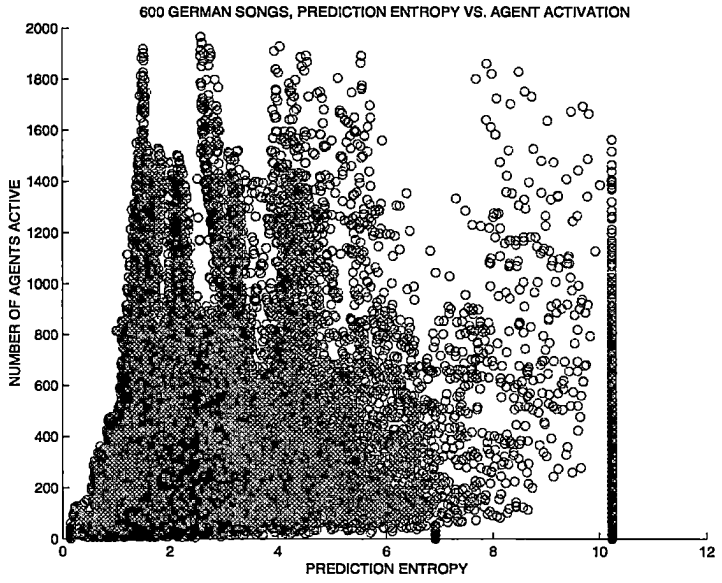


Figure 8.3: A scatter plot of prediction entropy against agent activation for 600 German folk songs.

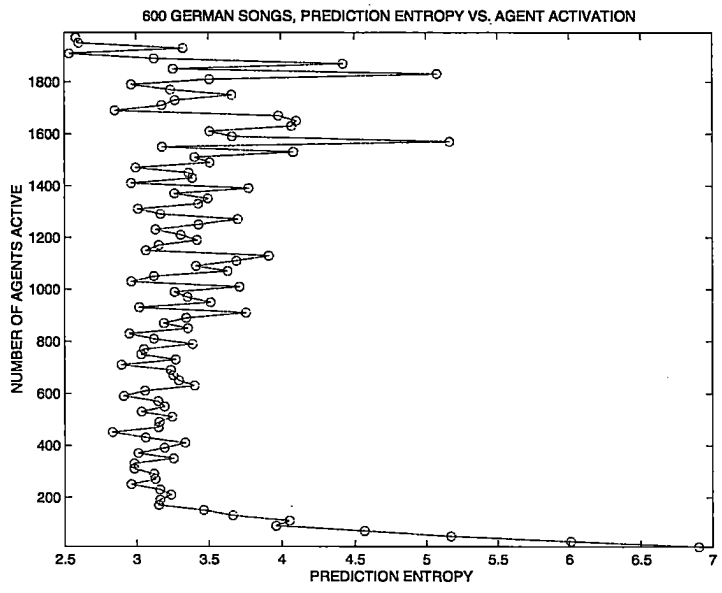


Figure 8.4: A slice-wise average of prediction entropy for different ranges of agent activation, derived from Figure 8.3.

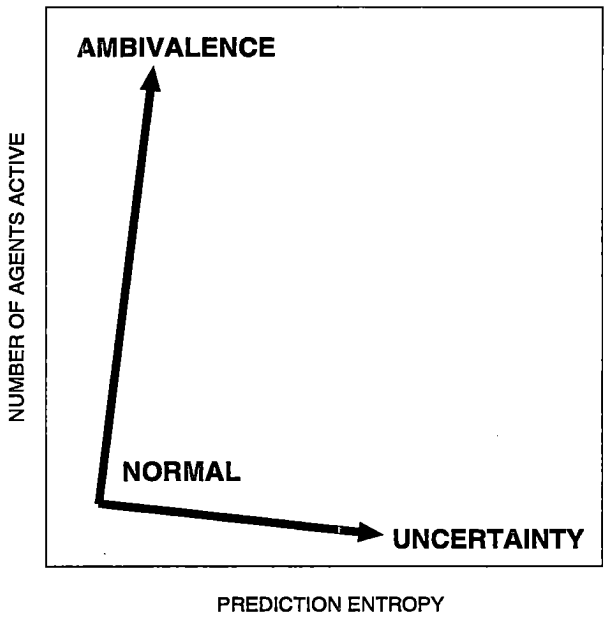


Figure 8.5: A schematic representation of the different types of ambiguity seen in Figure 8.4.

matically higher with decreasing agent activation.

These two trends are identified with the following two types of ambiguity, as shown schematically in Figure 8.5:

1. Ambivalence – "This has occurred many times before, and it could be any one of a number of things."

   The system has observed the current musical context many times before, and the current situation has been resolved in many different ways in the past. This situation leads to many agents being activated, generating different predictions. The allocation of probability density amongst the various predictions results in a somewhat flatter PDF, and the prediction entropy usually becomes *slightly* higher. Stated differently, since the probability is spread around to cover more possibilities, the correct answer ends up with less probability assigned to it. This effect is associated with the first trend above, where increasing numbers of agents leads to a slight increase in prediction entropy.

2. Uncertainty – "This has not occurred many times before, and there is little basis to know what to expect next."

   The system has observed the current musical context only a few times

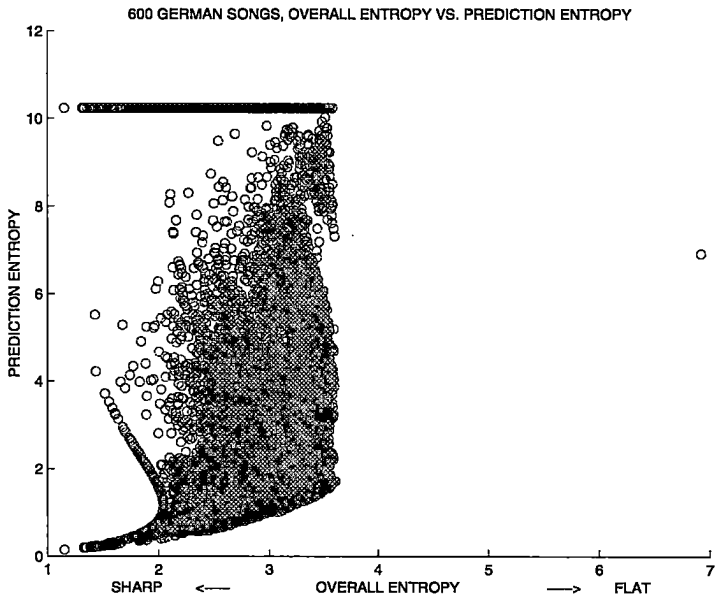


Figure 8.6: A scatter plot of overall entropy versus prediction entropy for 600 German folk songs.

before, if at all. Therefore, few agents are activated, and predictions are often based on an under-sampled model of the music. The poor quality of the predictions generated leads the prediction entropy to become *significantly* higher. This effect is associated with the second trend mentioned above, where very low numbers of activated agents lead to significant increases in prediction entropy.

Most of the data points in this experiment seem to be concentrated in the region labelled "NORMAL" in Figure 8.5. By analysing prediction entropy and agent activation, deviations from this region can be classified into the two different types of ambiguity described here.

8.3 Overall Entropy vs. Prediction Entropy

Having analysed each type of entropy with respect to agent activation, the relationship between the two types of entropy is now investigated. Figure 8.6 shows the overall entropy plotted versus the prediction entropy for the all the notes of the 600 folk songs.

The data at (6.92 , 6.92) results from flat distributions when no predictions are generated by Maestro. The points where prediction entropy is equal to 10.24 are those where predictions were generated, but none were correct.

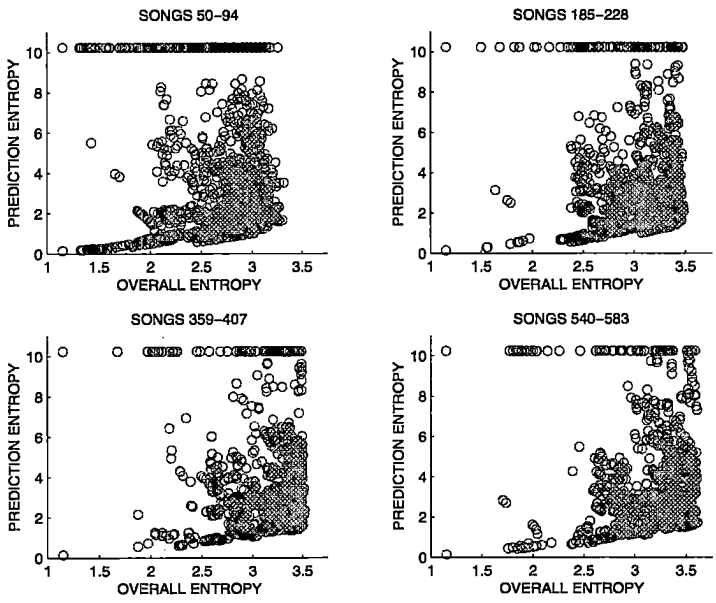


Figure 8.7: A scatter plot of overall entropy versus prediction entropy for 600 German folk songs, shown at different stages throughout the run.

The data in the figure shows a clear trend: for low levels of prediction entropy, (the bottom-most horizontal portion of the graph), the sharpest predictions achieve the best prediction performance (lowest overall entropy leads to lowest prediction entropy). Stated differently, in cases when Maestro's first choice prediction is correct, better results are achieved if other (incorrect) predictions are not given much weight in the PDF. This effect is fairly straightforward.

However, sharper predictions are not always beneficial. Figure 8.6 can also be analysed from the perspective of overall entropy. For low levels of overall entropy (the left-most vertical portion of the plot), a well defined limit-curve is evident. The very sharp PDFs represented in this portion of the graph are risky: if correct, they pay off (very low prediction entropy), but if wrong they can be disastrous (very high prediction entropy). For this range of values, there are no data points in the intermediate region, indicating that, as expected, very sharp predictions yield only extreme prediction entropy values, but not intermediate ones.

On the other hand, moving right along the graph, higher overall entropy means that the system is "playing it safe" with flatter PDFs, thus resulting in intermediate prediction entropy results. At these higher overall entropy values, data points are found along the full range of prediction entropy.

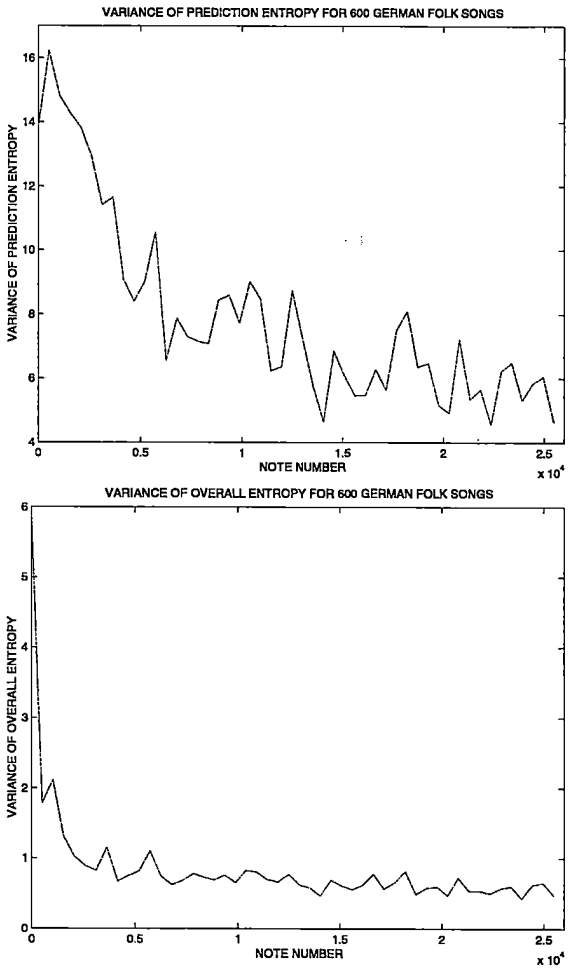


Figure 8.8: Variance of note-by-note prediction entropy (top) and overall entropy (bottom) during the course of listening to 600 German Folk songs (consisting of about 26,000 notes). The variance was calculated for 50 slices of 520 notes each.

The development of this relationship can also be studied over time. Figure 8.7 shows four scatter plots of overall entropy versus prediction entropy, sampled over certain time slices during training. From these plots it appears that the model undergoes gradual convergence. This is especially visible when comparing the left-most portions of the first and final graphs. With training, this dual-entropy characteristic of the model becomes more *focused* around a central operating regime.

To quantitatively measure this convergence, the variances of both the prediction entropy and overall entropy over time are plotted in Figure 8.8. A gradual and steady drop in variance for both types of entropy is clearly visible during the course of listening.

As Maestro learns, fewer risky predictions (both correct and incorrect ones) are made, replaced instead by more balanced predictions. It is proposed that the dual-entropy profile developed here can thus be used to measure the level of training, or *maturity* of a model. Untrained models make risky predictions that result in extreme prediction performance levels, while more mature models make fewer risky predictions, leading to more moderate values of prediction entropy.

This effect clearly depends on the nature of the corpus used for training. A very homogeneous corpus containing many instances of a few repeating patterns would lead to sharp probability distributions that do not flatten out with time. However, with a large realistic corpus, this is not expected to be the case.

8.4 Summary

This chapter described experiments performed to study ambiguity. Two types of entropy and a measure of agent activation are used.

Overall entropy is compared with agent activation, revealing that with training, more agents are instantiated, generating more predictions, and flattening the prediction PDFs. This flattening reaches a point of saturation likely due to the homogeneous nature of the training corpus.

A plot comparing prediction entropy with agent activation is shown to identify two types of ambiguity: ambivalence and uncertainty. Most of the data points are found in a central operating regime, deviations from which can be classified as different types of ambiguity.

The entropy characteristics of the model are shown to develop with experience. A dual-entropy profile is shown to provide a measure of the level of training, or maturity of a model. With experience the model generates less risky predictions and achieves more consistent prediction entropy results that are better overall.

Chapter 9

Validating PGS

As described in Chapter 3, the design of Maestro's segmentation stage is based on the theory that *Perceptually Guided Segmentation* (PGS) leads to more efficient models for prediction than other segmentation strategies. This chapter reports on experiments performed to empirically validate this theory.

After reviewing and formalising the theory behind PGS, an experimental method developed to verify this theory is described. The experimental results from using this method are then reported and analysed.[1]

9.1 Formalising PGS

Prediction using context models relies on repetition in the data – once a pattern is seen, it can be predicted correctly if seen again. Repetitions in music are often re-appearances of structurally salient patterns in the piece. It is therefore conjectured that the most useful segments to store for prediction would be those that correspond to these structurally salient patterns; since they are correlated to the intrinsic structure of the music, these segments would be more likely to reappear in-full later on in the piece. Conversely, segments that do not lie on these structural boundaries are less likely to reappear in full, and are therefore less likely to be useful for prediction. If the more useful segments are kept, and the less useful ones are ignored, it is argued that the overall efficiency of the model for prediction, considering both model size and robustness, is improved.

Recall from the discussion in Chapter 3 that there is a trade-off inherent to fixed-order context models: higher-order models generally perform

[1] An earlier version of this work appears in [94]. The results reported in [94] vary from the present research in that a different method was used when combining the probability distributions generated by Maestro's prediction stage. The results reported in this chapter are consistent with the implementation described in the rest of the dissertation.

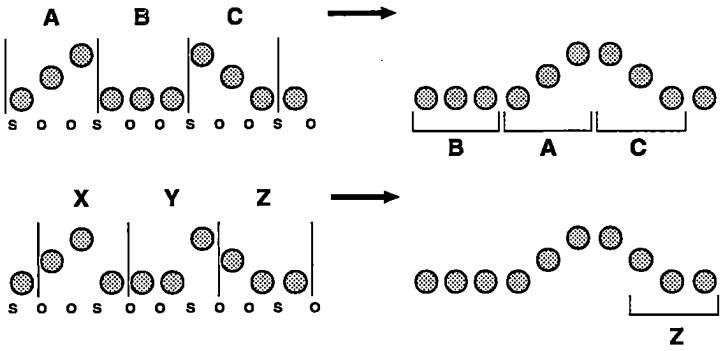


Figure 9.1: Part of the theory underlying perceptually guided segmentation: Segmentation according to s-points (top) leads to models with better prediction performance than segmentation according to o-points (bottom). The model at the top captures the salient patterns present in the music, and is therefore able to spot repetitions even when they appear in a different order. Conversely, the model resulting from the bottom segmentation cannot.

better, while lower-order models are smaller and easier to train. Maestro's approach attempts to draw a compromise between them by storing segments of various lengths, as a result of the perceptually guided segmentation. This way, more costly, longer segments are only stored in the model when it is appropriate and beneficial to do so. Furthermore, it is argued that these salient patterns are more appropriate for use in parsing the music. Finally, using this approach, learning and listening truly arise from one and the same process.

The concept of perceptually guided segmentation can be formalised into the following two hypotheses:

1. It is hypothesised that there exist certain points of segmentation in a piece, named here *s-points*, which lead to a context model with better prediction performance. Other points, *o-points*, lead to context models with worse prediction performance (Figure 9.1).

2. It is further hypothesised that the segmentations suggested by a perceptually guided segmentation strategy such as Maestro's PGS correspond to the s-points in the music.

The goal of the experiments reported in this chapter is to validate the existence of the concept of s-points. It may very well be that there is, in fact, no systematic correlation between the points at which the music is segmented and the efficiency of the resulting model for prediction. If, however, the reverse is true and s-points do indeed exist, the experiments have a further aim of testing whether or not Maestro's PGS strategy – and by extension other similar perceptually-based segmentation strategies – is correlated with the s-points in the music.

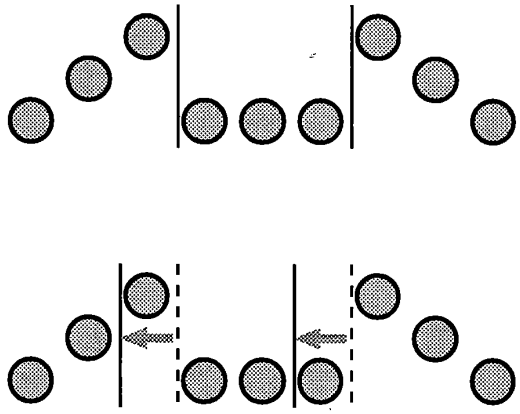


Figure 9.2: N-Note Segmentation Shifting. In the example shown, the segmentation suggested by PGS (top) is shifted back by one note (bottom) and then stored in the model.

9.2 Method: N-Note Segmentation Shifting

In order to test the above hypotheses, the prediction performance of a system based on PGS needs to be compared with systems based on different segmentation and modelling techniques. Ideally, one would want to compare PGS with other, non-perceptually guided segmentation strategies from the literature such as Conklin and Witten's, or with a totally random segmentation strategy.

Unfortunately, a complication arises: other strategies result in models with different numbers of segments and different distributions of segment lengths. Since both of these factors affect prediction performance, a proper systematic comparison cannot be made with the PGS model to judge efficiency: a large inefficient model might easily outperform a smaller, more efficient model. Thus, a truly fair comparison would involve two models of a similar size and similar distribution of segment lengths[2].

In order to address these issues, an experimental method called *N-Note Segmentation Shifting* (NNSS) is developed. As the name implies, whenever a specific segment is suggested by the PGS strategy for storage in the model, the system stores a segment of the suggested length, *but shifted back by n notes* (Figure 9.2). This leads to the same number of segments and same

[2]One method of achieving this would be to measure how large the PGS model gets, and set this as the upper limit on the random-segmentation model size. Unfortunately, this too causes difficulties, since in trials a random model reaches this upper limit well before the corpus is fully processed. The random segmentation model is then only trained on a narrower data set, and again, a proper comparison cannot be made.

segment length distribution being suggested to each model, as well as both models drawing segments from the same range of training data. Slight differences in model sizes can result if one segmentation strategy leads to more repeated patterns being suggested, and thus one model can end up slightly smaller. However, model efficiencies can still be compared, as described below.

How would the results of NNSS relate to the PGS hypotheses? If both of the hypotheses mentioned in the previous section are correct, PGS would suggest segmentations aligned with the s-points in the music. The delineated segments would thus be correlated with the underlying structure of the piece, and would therefore be more likely to repeat later in the music. Therefore, the PGS strategy should lead to a model with better prediction performance than a model produced by a shifted-PGS strategy (NNSS). Additionally, since PGS would capture more repetitions, the total model size should be smaller. These are called here the *direct effects* of PGS.

If, however, the second hypothesis is false, then the PGS model should not produce significantly better prediction performance, as it would not take advantage of the s-points in the data. Furthermore, if both of the hypotheses are false and there are no such entities as s-points, then no significant corresponding trend should appear in the data.

Another set of observations can be made. If PGS theory is correct, smaller shifts should have more drastic effects than larger shifts. When beginning at an s-point, a small shift (of say, one note) makes it unlikely that another s-point might be reached. Therefore, the system will most likely store all its segments beginning and ending on o-points, and prediction is expected to be significantly worse. On the other hand, when shifting a longer distance away from an s-point, while it is still possible that an o-point will be reached, it is more likely than before that another s-point will be reached. This more even mix of s-points and o-points would lead to a somewhat intermediate prediction performance. In general, the larger the shift distance used, the smaller the phase correlation between the starting s-point and the point reached by the shift, and thus the less drastic the effect of the shift is expected to be. This is called *the PGS phase effect*, and is used along with the direct effects of PGS to analyse the results of the following NNSS experiments.

It is important to emphasise that the main aim here is to test whether or not there exist optimal points for segmenting music for the purpose of prediction (s-points). From this perspective, NNSS can be seen as a perturbation test: by perturbing the segmentation strategy from the original setting, one examines whether or not the original setting is an optimal one. If performance degrades with perturbation, the original setting is at least a relative optimal point and the hypothesis is supported. However, if no clear correlation between perturbation and performance is observed, then the original setting cannot be said to be a relative optimal point and the hypothesis is not supported.

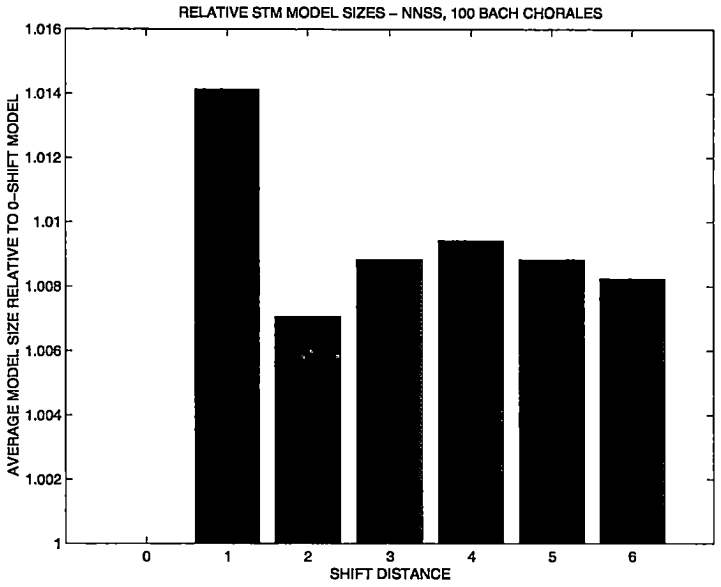


Figure 9.3: The STM context model sizes resulting from different shifts, averaged over the 100 Chorales, relative to the PGS (zero-shift) model size.

9.3 Experiments: Short Term Memory

The repetition of certain segments of notes is more pronounced within a single piece than across a musical style. Therefore, in order to better observe the potential effects of PGS, the STM model is examined first.

The 100 Bach chorales used by Conklin and Witten were presented to Maestro, and the usual perceptually-guided segmentation strategy was used. The experiment was then re-run another six times on the same 100 Chorales, but these times with increasing segmentation shifts of one through six notes. (For these experiments, Maestro was configured to read in the shift parameter at run-time.)

The results from this experiment are analysed using the framework described in Chapter 7 for studying music learning: model growth, the number of predictions generated, and the prediction performance.

Figure 9.3 shows the relative sizes of the STM context models resulting from the various segmentations, averaged over the 100 Chorales. The following observations can be made:

- PGS results in by far the smallest model size. This is a result of PGS picking up the most repetitions, and is consistent with both hypotheses stated above.

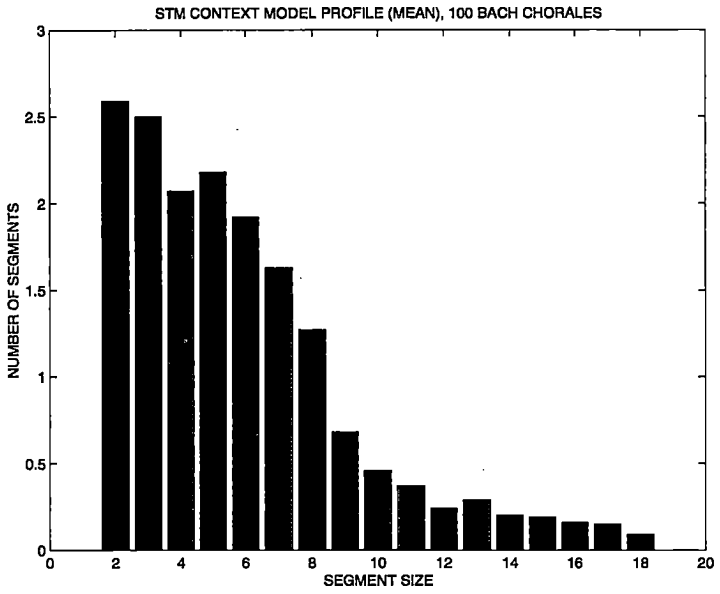


Figure 9.4: Average STM context model profile for the 100 Chorales.

- While all shifts lead to larger model sizes, the smallest shift has the most significant effect, resulting in the largest model. This result is consistent with the PGS phase effect.

The average model size histogram of the zero-shift STM context model for the 100 chorales is shown for the various segment lengths in Figure 9.4. For the STM models, there are many segments of length two. Therefore a shift of one note should be very significant, while greater shifts should, on the whole, produce less drastic effects. This is because, according to PGS theory, larger shifts may well lead to other s-points. The data is consistent with this theory.

Moving on to the next method of analysis, Figure 9.5 shows the relative number of predictions generated from the models resulting from the various segmentations.

- The PGS model makes the most predictions, *despite having the smallest model size.* This otherwise unlikely result is consistent with the idea that PGS results in more efficient models for prediction.

- The model resulting from a shift of one, despite having the largest model size, makes the fewest predictions. This result is also in stark contrast to what would otherwise be expected, and is strongly consistent with the PGS phase effect.

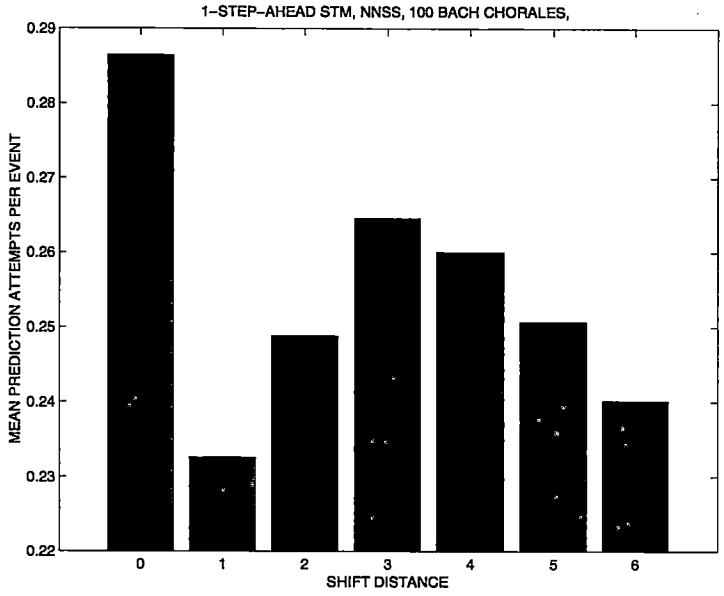


Figure 9.5: Average number of STM 1-step-ahead predictions made as a result of different shifts for 100 Bach Chorales.

These two observations provide strong support that PGS leads to more efficient models for prediction. Why else would the smallest model generate the most predictions and the largest model the fewest?

Finally, Figure 9.6 shows the relative levels of prediction entropy resulting from the various segmentations.

- The PGS model has the best prediction performance, despite having the smallest model. This is consistent with PGS theory.
- The models resulting from shifts two and three make the worst predictions. This is not fully consistent with the PGS phase effect, which would have predicted that the one-shift model would make the worst predictions.

On all three counts mentioned above, the results are consistent with the direct effects of PGS – the PGS-based model consistently performs best, despite having a smaller model size. As stated above, these results would be hard to explain otherwise. Additionally, on two of the three counts the results are consistent with the PGS phase effect, and the trends are as expected. Therefore, the STM experiments provide strong support for the PGS hypotheses. The evidence suggests that there do indeed exist optimal points for segmenting music for the purposes of prediction, and that these points are correlated with perceptual segmentation cues in the music.

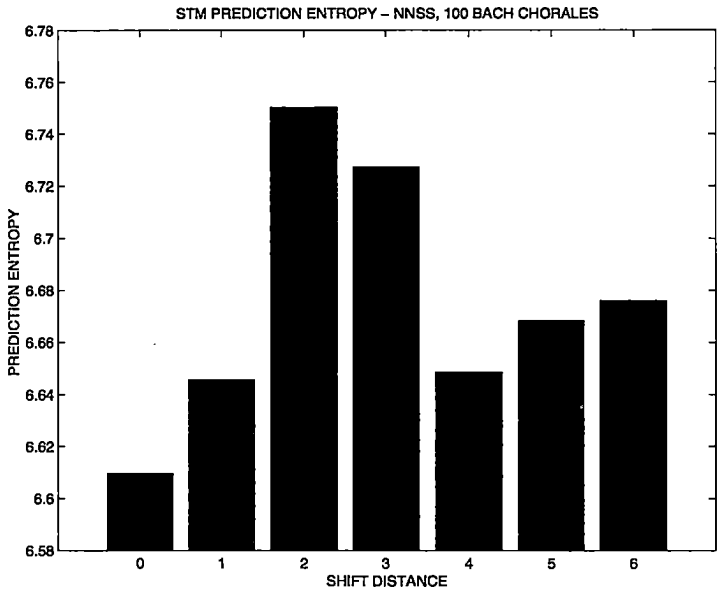


Figure 9.6: STM 1-step-ahead average prediction entropy resulting from different shifts.

9.4 Experiments: Long Term Memory

Unlike short term memory, long term memory stores information from one song to the next, and it is not necessarily the case that storing specific patterns from one song will be beneficial in dealing with patterns from other songs as well. To test if PGS theory applies to extra-opus information, the LTM context models were examined for the same experiments performed above with the STM context models.

Figure 9.7 shows the relative sizes of the context models resulting from the various segmentations.

- PGS results in by far the smallest model size. This is caused by PGS picking up the most repetitions, and is consistent with both hypotheses.
- While all shifts lead to larger model sizes, the smaller shifts as a whole have slightly more significant effects, resulting in larger models. This is consistent with the PGS phase effect, but is not as pronounced as in the STM results.

Figure 9.8 shows the histogram of the LTM context model at the end of listening to 100 chorales. The lower end of the histogram is shaped differently than in the STM histogram. Unlike STM, the LTM context

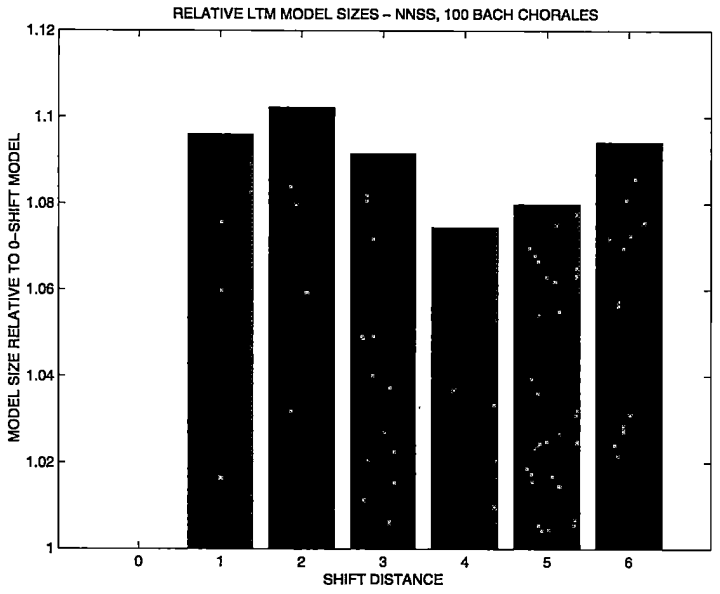


Figure 9.7: Relative LTM context model sizes resulting from different shifts.

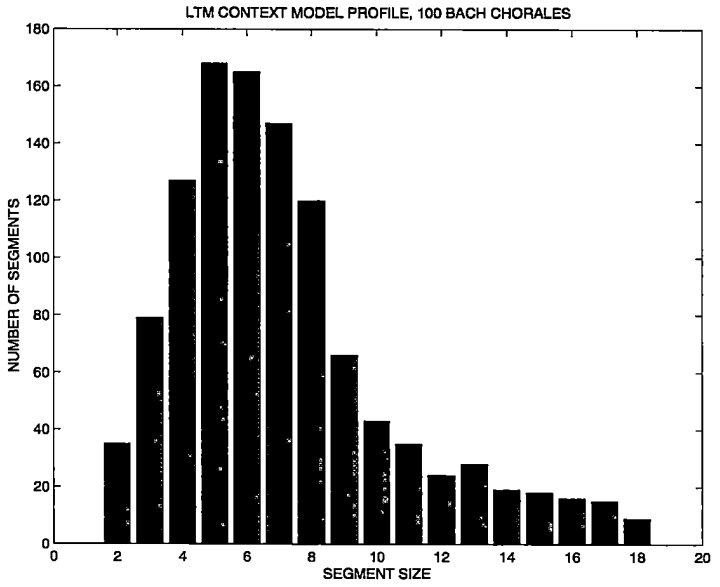


Figure 9.8: Final-state LTM context model profile.

model is not reset at the end of each piece. Therefore, while a segment that often repeats is incorporated into numerous STM context models (and therefore contributes repeatedly to the average profile), it is only added once to the LTM model. This comparison indicates that shorter segments are recommended for inclusion in the model more often than longer ones. It is only the smaller number of combinatorially possible short segments that keeps down their number in long term memory.

For the LTM model, there are many segments of length four through eight. Therefore small shifts of one to three notes should be more significant, while greater shifts should on the whole produce less drastic effects. This is consistent with the data.

The significantly larger sizes of the shifted-PGS models are highly supportive of PGS theory: o-points lead to fewer repetitions being captured, and thus larger model sizes. Note that the difference in LTM model sizes seen here between PGS and the shifted models (roughly 8 - 10 percent) is an order of magnitude larger than the difference seen in the STM model sizes shown in Figure 9.3 (roughly 0.8 - 1.4 percent). This is a crucial point.

Over the course of listening, a shifted model accumulates many non-ideal segments, namely, those aligned with o-points. In small numbers (i.e. in STM) these do not add much to the predictive abilities of the model. However, the LTM model stores them permanently, and over time, the number of extra non-ideal segments in LTM becomes very large (an order of magnitude greater in LTM than in STM). Enough residual information is contained in these non-ideal segments to prove useful in the long run, and the shifted models can outperform the PGS model simply due to their much larger size.

Recall however, that the theory being tested is that PGS leads to models that are *more efficient* for the purpose of prediction. The fact that much larger models perform better than PGS is by no means inconsistent with the PGS hypotheses. Relative efficiency can only be compared between models of similar size. Brute size can win out over a more efficient, but much smaller model. With this in mind, the remaining results are analysed.

Figure 9.9 shows the relative number of predictions generated from the models resulting from the various segmentations:

- PGS leads to the fewest predictions being made. This is likely the result of the great difference in model sizes, which serves to override any of the effects that PGS may or may not have, as explained above.
- The models resulting from the smaller shifts make fewer predictions, despite having larger model sizes. This result is consistent with the PGS phase effect, although not as pronounced as in STM.

Figure 9.10 shows the relative levels of prediction entropy resulting from the various segmentations:

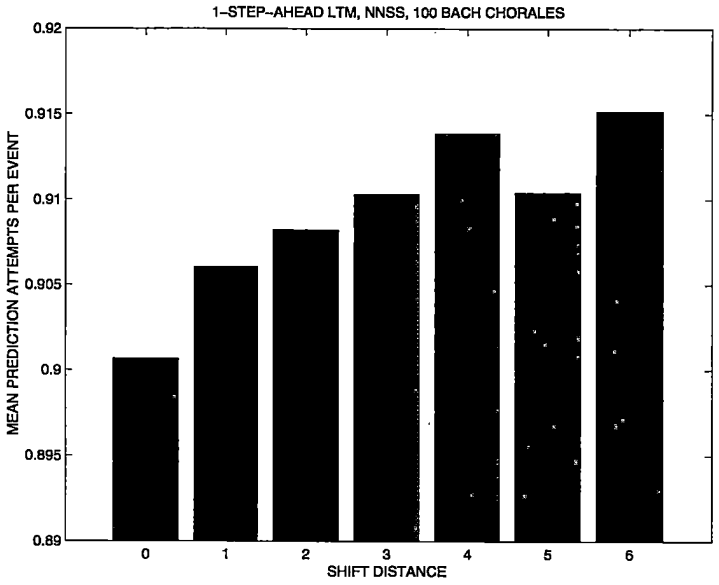


Figure 9.9: Average number of LTM 1-step-ahead predictions made as a result of different shifts.

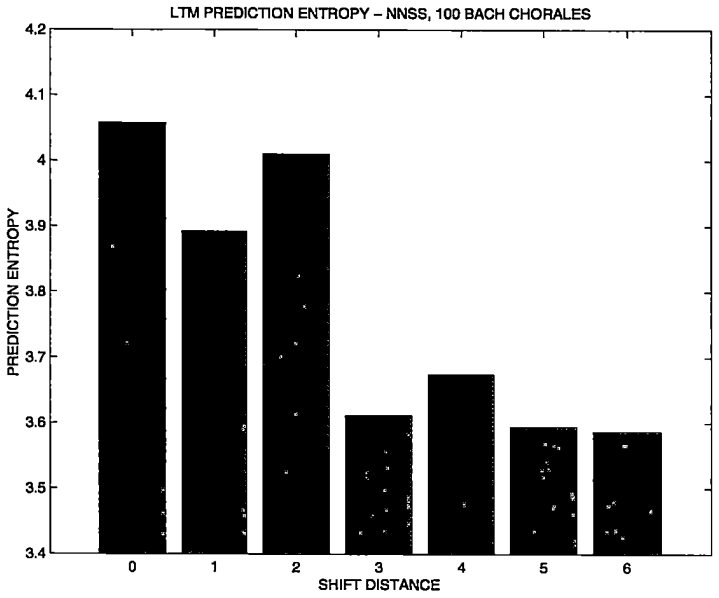


Figure 9.10: LTM 1-step-ahead prediction entropy resulting from different shifts.

- PGS leads to the worst predictions. Again, this is likely due to the overwhelming difference in model sizes.
- The models resulting from smaller shifts on the whole make worse predictions than those resulting from larger shifts, despite having larger models. This is consistent with the PGS phase effect.

The large difference in model sizes and the presence of the PGS phase-effect suggests that PGS is playing a role in LTM, but that the difference in model sizes is simply too great to allow detection of the two other direct effects of PGS (number of predictions and prediction entropy).

The experimental work cited by Sloboda in Section 3.2.5 suggests that humans store melodic information in segments delineated by perceptual cues. The results presented here provide the first empirical evidence suggesting that *this strategy is actually more efficient for prediction*, and thus constitute a major contribution of this research.

With regard to long term memory, an additional point needs to be considered. In Maestro, once information is stored in short term memory, it is rolled over into long term memory at the end of the piece. This flow from one memory to another seems more likely in humans than there being an independent segmentation strategy for each one. Therefore if PGS is likely to be present in STM, as the results suggest, the same model would likely be rolled over into LTM.

9.5 Summary

An experimental method named N-Note Segmentation Shifting is developed to verify the hypotheses underlying the theory of Perceptually Guided Segmentation.

Experiments with short term memory are highly consistent with PGS theory, and some of the more unlikely results are difficult to explain otherwise. Experiments with long term memory are still consistent with parts of PGS theory, but the large differences in model sizes prevents a fair comparison of efficiency from being made.

Chapter 10

Multi-Style Results

Whereas the previous three chapters presented experiments performed with music from a single musical style, this chapter presents experiments performed with music from multiple styles. First, the Style Switching and Comparative Listening experiments proposed in Chapter 1 are carried out and the results are analysed and discussed. Then, a series of Geographical Mapping experiments are performed, taking advantage of the capabilities of Maestro, as well as the great diversity of music present in the Essen Folk Song Collection.

10.1 Style Switching and Comparative Listening

As described in Section 1.3.5, Style Switching involves training a listener in one style and studying how that listener responds to another. Comparative Listening involves presenting the same piece to two listeners who have different musical backgrounds.

To re-invoke the twin analogy introduced at the beginning of this dissertation, consider two twin brothers separated at birth and brought up in different countries – one in China (call him C), the other in Germany (G). At a certain stage of development, the brothers are reunited in Germany and C moves in with G. Two specific questions can be asked: How will C deal with his new musical surroundings (Style Switching)? Also, what advantage will G have over C, having grown up in the native German musical culture (Comparative Listening)?

To investigate these issues, two fresh instantiations of Maestro (for the sake of simplicity, hereafter referred to as 'listeners') were presented with the data sets shown in Table 10.1. Music from German and Chinese styles was chosen for training the two instantiations of the system, as these styles are known to be different from one another, and are also abundant in the

Listener	Phase I	Phase II
C	600 Chinese songs	600 German songs
G	600 German songs	600 German songs

Table 10.1: Data sets used for the Style Switching and Comparative Listening experiments.

EFSC data sets available.

In one of the few articles published about the EFSC, the late Helmut Schaffrath notes some of the significant differences between the German and Chinese folk songs. He notes that:

1. German folk songs generally skip more often up than down; in the Chinese pool the opposite applies.

2. Chinese songs use descending intervals of a third 45% more often than German songs do, and this might be explained by the preference for pentatonic scales.

3. German songs use the interval of a fourth 51% more often than Chinese songs do. This may be explained by the preference for tonic and dominant scale degrees in the European tonal system and by the high incidence of upbeats in German folk songs [106, pp. 107-8].

Due to these differences, these two groups of songs were selected for use in the present research. The Chinese songs used have EFSC reference numbers `han0001-han0600`, and the groups of German songs used for Phase I have reference numbers `deut0567-deut1166` and `deut1167-deut1766` for Phase II.

The same framework developed in the previous chapters to analyse long-term music learning is also used here to investigate the results of multi-style experiments: context model growth, number of predictions generated, and prediction performance.

10.1.1 Context Model Growth

To begin, the context model growth rate of both listeners is examined (Figure 10.1). Both listeners independently show model growth gradually slowing down in Phase I. Since different data is being processed by each one, the graphs are uncorrelated.

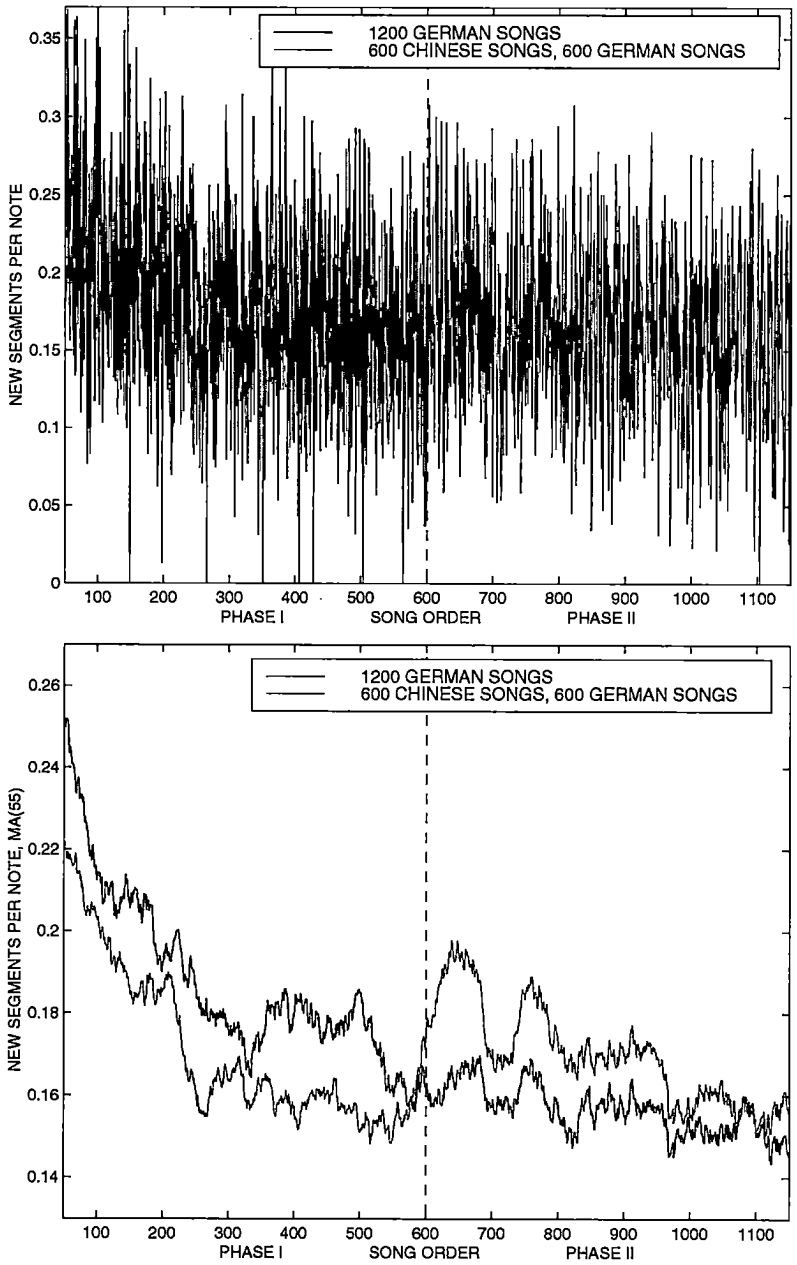


Figure 10.1: Context model growth rate per note-event for C and G (top), also shown smoothed to reveal the trends (bottom).

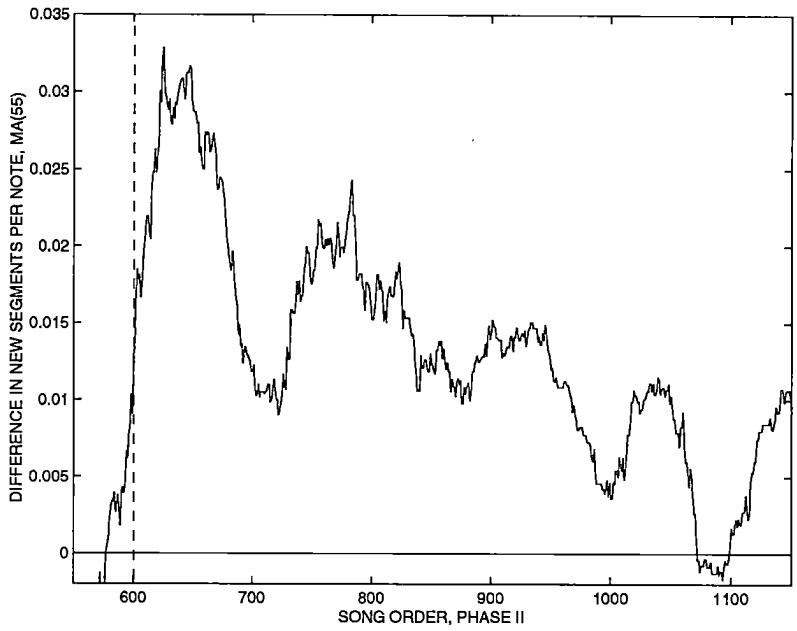


Figure 10.2: Difference in context model growth rate between C and G during Phase II, smoothed with a moving-average to better reveal the trend.

With the onset of Phase II, the model growth of C suddenly increases to deal with the new, as yet unseen, information. Meanwhile, G continues its slow gradual decrease in context model growth, since from G's perspective the music has not changed significantly from earlier experiences.

In Phase II the two plots are clearly correlated, as both listeners are now listening to the same music. While the two listeners have different overall levels of competence, the songs themselves display an inherent entropy of their own that affects both listeners in a similar way.

Additionally, during Phase II, C begins to learn the German style and the responses of the two listeners gradually converge with experience. Figure 10.2 shows the difference between the context model growth rates of C and G during Phase II, highlighting this convergence.

10.1.2 Number of Predictions Made

Next, the number of predictions generated by each listener is analysed. Figure 10.3 shows that for both C and G, the number of predictions gradually increases for all horizons during Phase I. Again, the graphs are uncorrelated since each listener is processing different pieces of music. With the onset of Phase II, the number of predictions generated by C suddenly drops since many of the musical contexts from the new style are unfamiliar. On the other

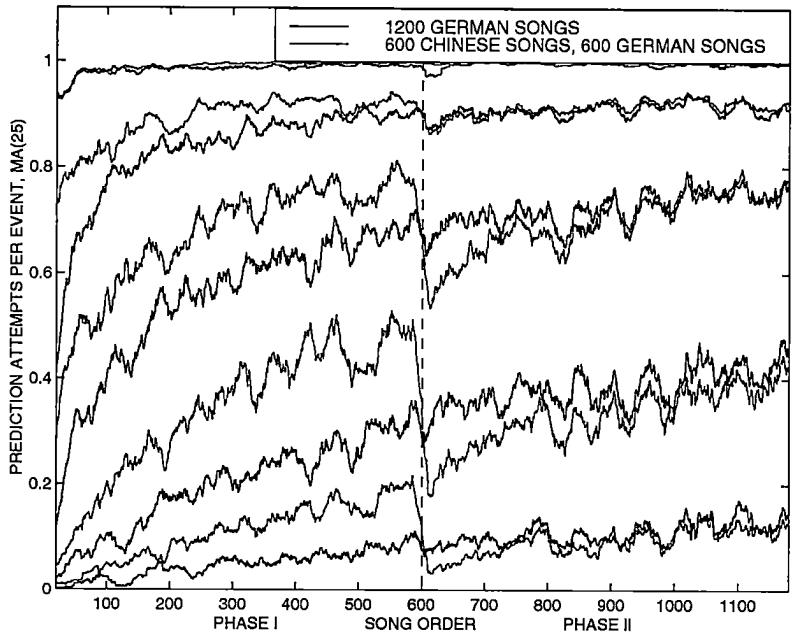


Figure 10.3: Number of predictions generated by C and G for forecast horizons of (top to bottom) one through five steps ahead.

hand, G continues the gradual increase in the number of predictions made, as no significant change has occurred in the music from G's perspective.

Figure 10.4 shows the difference in the number of predictions generated by C and G for various forecast horizons during Phase II. With experience, C learns the new style and the responses of the two listeners gradually converge.

Figure 10.5 shows the data for C alone, to better highlight the sudden drop in the number of predictions brought on by the change in style between Phase I and Phase II.

10.1.3 Prediction Performance

Finally, in Figure 10.6 the prediction performance is examined using prediction entropy as a measure. Both C and G independently show the prediction quality gradually improving for all horizons during Phase I. The graphs are uncorrelated since different data is being processed by each listener.

With the onset of Phase II, C's prediction performance suddenly deteriorates since C does not recognise the music from the new, as yet unseen, style. Meanwhile, G continues the gradual improvement in prediction performance.

Figure 10.7 shows the difference in prediction entropy between C and G

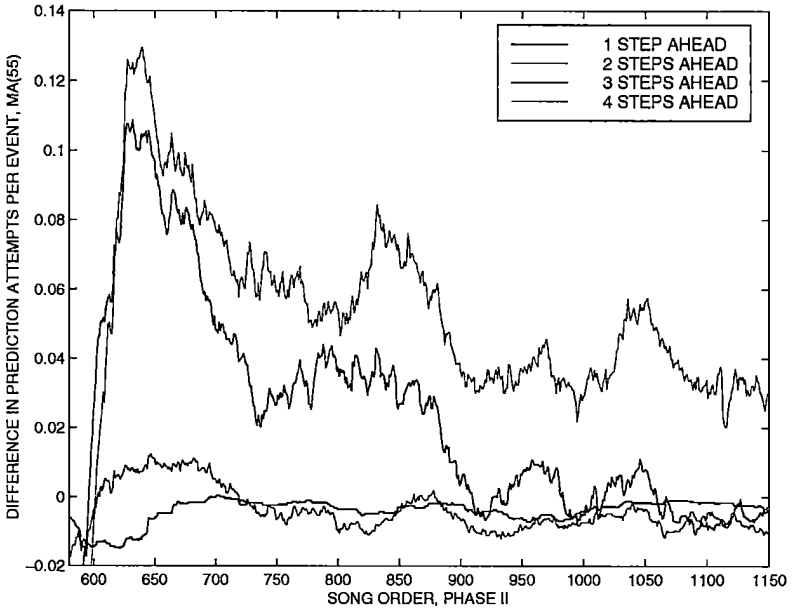


Figure 10.4: Difference in number of predictions made between C and G during Phase II, smoothed with a moving-average to better reveal the trends.

for various forecast horizons during Phase II. With experience, C learns the new style, and the performance levels of the two listeners gradually converge.

Figure 10.8 shows C's prediction entropy alone to better highlight the sudden degradation in prediction performance brought on by the change in style.

10.1.4 Discussion

These results are revealing about the nature of the responses brought on by a drastic change of musical style. From the perspective of Style Switching (looking only at listener C), a change in style causes three drastic changes: the context model growth rate increases, the number of predictions generated drops and the prediction entropy increases. All these effects then gradually diminish as the new style is learned.

From the perspective of Comparative Listening, both C and G are analysed during Phase II. The results show that as expected, previous experience with the musical style proves to be an advantage for the native listener over the foreign listener. However, even though each listener has an independent level of competence, the results of the two are still correlated due to the intrinsic entropy of the individual pieces. With experience, the foreign listener (C) learns the new style, and the difference in performance level between the two listeners gradually diminishes.

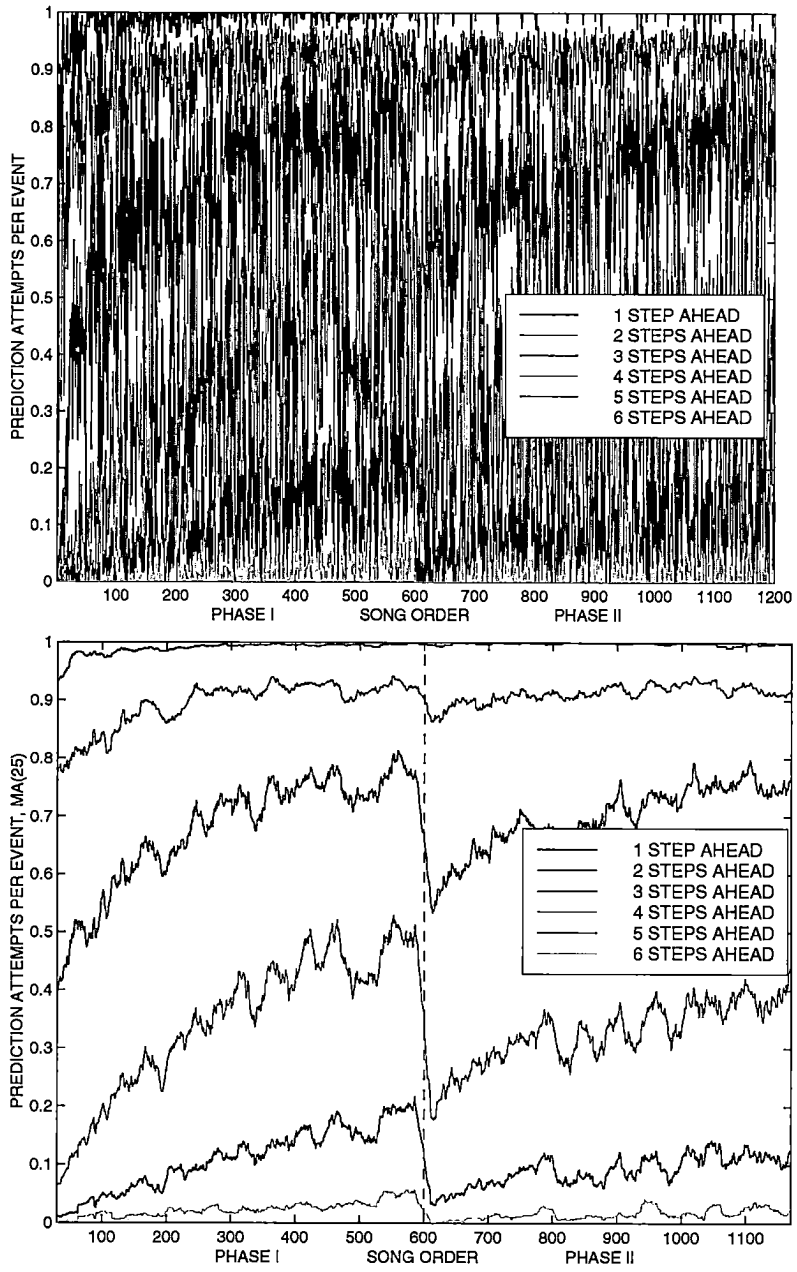


Figure 10.5: Number of predictions of various orders made by C (top), also shown smoothed with a moving-average to better reveal the trend (bottom).

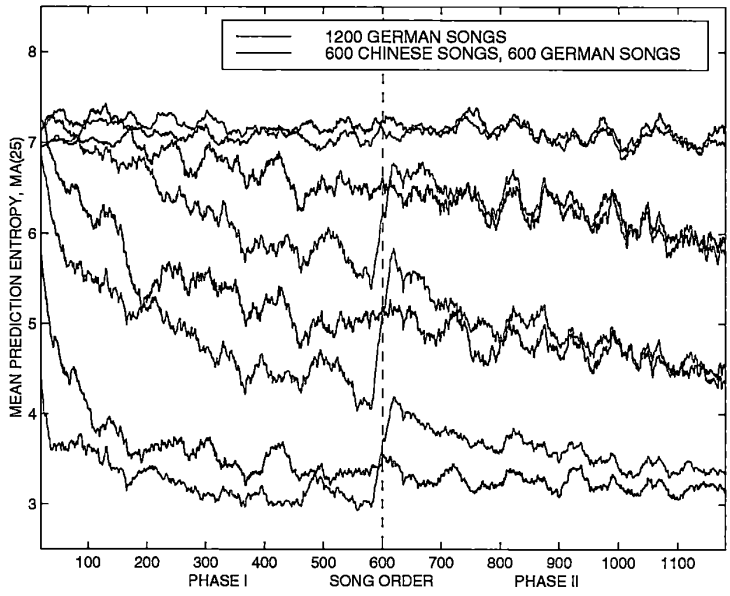


Figure 10.6: Prediction entropy for both C and G shown for forecast horizons (bottom to top) one through four steps ahead.

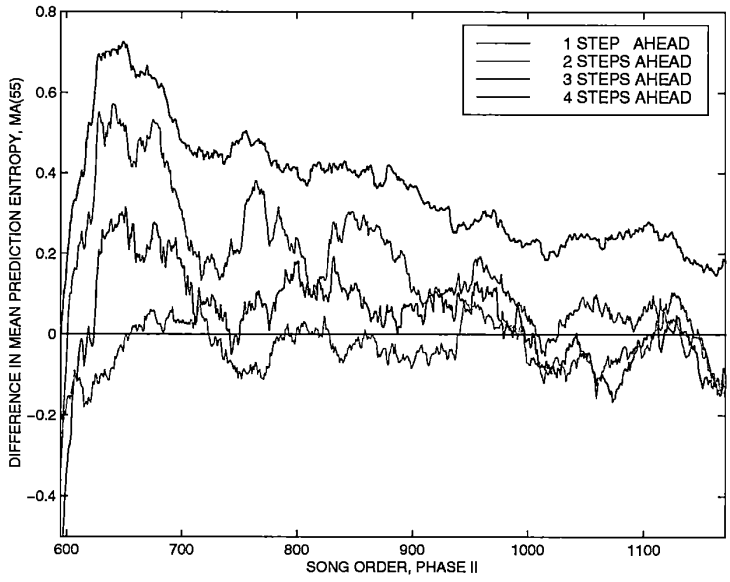


Figure 10.7: Difference in prediction entropy between C and G during Phase II, smoothed with a moving-average to better reveal the trend.

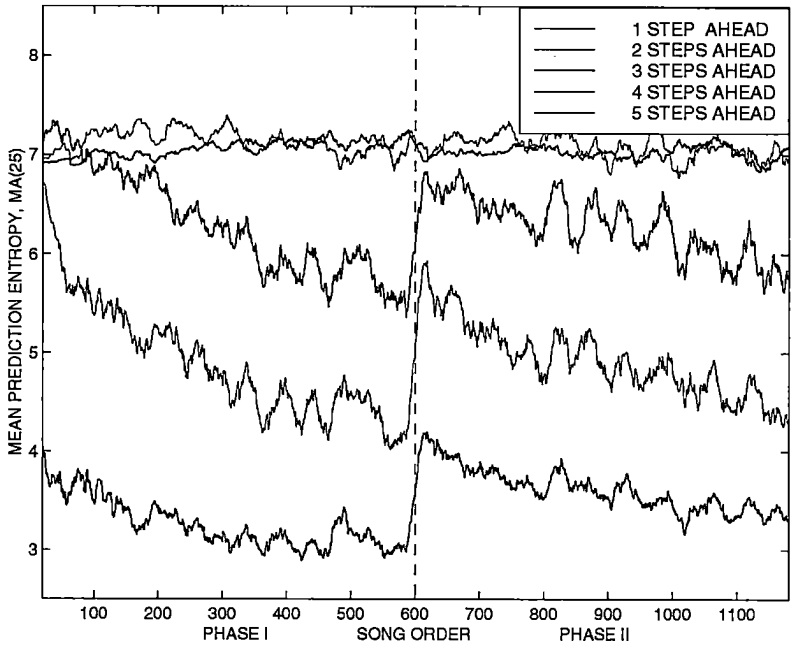


Figure 10.8: C's prediction entropy for various forecast horizons, smoothed with a moving-average to better reveal the trend.

As mentioned in Section 4.4.1.2, the concept of forgetting is not implemented in Maestro's long-term memory. Forgetting would have a significant effect on the above experiments, as the foreign listener would be quicker to adapt to the new style if biases from previous experiences diminished with time. While it is likely that forgetting plays a role in people's long term musical memories, there is no clear value known for the rate of decay involved. Therefore, since setting the decay rate to an arbitrary value would affect the results arbitrarily, it was considered best simply not to implement forgetting, and then interpret the results with this qualification in mind.

Style Switching and Comparative Listening experiments have been performed and the results analysed. Consider now what would happen if, in the above experiments, the second musical style were not so drastically different from the first. Would the dramatic responses seen in C's results be attenuated? How would the relative geographical proximity of the origin of the two musical styles affect the magnitude of the responses? Drawing on the wide variety of music available in the Essen Folk Song Collection, Maestro is configured to perform one final set of experiments to address these questions.

10.2 Geographical Mapping

Returning one last time to the musical twins analogy, consider what would happen if, instead of C moving to Germany, both brothers embarked on a world-wide tour. Having grown up in two significantly different musical cultures, each claimed that his native musical upbringing had better prepared him for listening to the music of the world's different cultures. To settle the score, C and G decided to try their hand at predicting music from different parts of the world over the course of their travels.

To perform this Geographical Mapping experiment, each instantiation of the system was trained on 600 songs of the respective native style, exactly as in Phase I above. Then, sets of songs from 33 different styles, originating in 28 different countries were presented to both systems and the average prediction entropy was calculated for each style. The EFSC reference numbers of songs used in this experiment are given in Appendix D.

For these experiments, learning was switched off after the first 600 songs to ensure that the system was not affected by any music other than the original country of training. As a result of this, the order of presentation of the other musical styles does not matter. The results of the experiment are given in Table 10.2.

Different numbers of songs were available for the various countries, as indicated in the 'SAMPLES' column. In cases where a large number of songs were available, a maximum of 50 were used. The results from multiple styles originating in a single country were combined in a weighted average to arrive at an overall prediction entropy value for that country.

Before analysing the results, the observations made here must be qualified by the extent to which the samples used are representative of the music from the specific region. Krumhansl [68] reports that excerpts from different musical styles can reflect actual stylistic differences or simply differences in the specific excerpts. Similarly, Meyer in the book *Style and Music* notes that the probability relationships involved in understanding a piece of music depend upon the breadth of the sample on which probability estimates are based [80, p. 61]. Westhead and Smaill also report that good style-discrimination performance is achieved with their system when a large enough number of pieces is used for training [125]. This qualification is related to the general statistical consideration of small-sample variability. While the qualification might be relevant for countries with few samples, the data sets containing 50 samples are likely representative. With this in mind, the results are now analysed.

As an initial sanity check, it is encouraging to note that each listener performs significantly better on music from its native culture. Additionally, multiple styles originating in the same country yield similar prediction

COUNTRY	*SAMPLES*	*CHINESE*	*GERMAN*	*DIFFERENCE*
Austria	50	4.65	4.00	0.65
Brazil	1	2.91	3.08	-0.17
Canada	1	4.21	3.53	0.68
China - Han	50	3.01	3.96	-0.95
China - Natmin	50	3.42	4.02	-0.60
China - Sinaxi	50	3.11	3.84	-0.73
China - Xinuha	10	3.45	4.04	-0.59
China	160	3.20	3.94	-0.75
Czech Republic	43	3.95	3.56	0.39
Denmark	9	4.11	3.63	0.48
France - General	14	4.14	3.63	0.52
France - Elsass	50	4.02	3.37	0.65
France	64	4.05	3.43	0.62
Germany	50	4.10	3.40	0.70
Hungary	45	4.02	3.55	0.47
India	1	2.94	3.03	-0.09
Italy - Tirol	8	4.20	3.70	0.50
Italy - General	14	5.12	4.63	0.48
Italy	22	4.78	4.29	0.49
Japan	1	5.54	4.71	0.83
Java	1	4.92	4.00	0.92
Luxembourg	8	3.60	2.99	0.61
Mexico	4	3.82	3.32	0.51
Netherlands	50	3.73	3.20	0.53
Poland	25	4.12	3.58	0.54
Romania	28	3.83	3.46	0.38
Russia	37	3.82	3.37	0.44
Saudi Arabia	1	4.34	3.39	0.95
Sweden	11	4.01	3.39	0.63
Switzerland	50	4.32	3.67	0.66
Syria	1	3.34	2.77	0.57
Turkey	1	3.14	3.03	0.11
Ukraine	13	3.95	3.54	0.40
United Kingdom	4	4.19	3.35	0.84
USA	7	3.56	3.60	-0.05
Yugoslavia	50	4.05	3.32	0.73

Table 10.2: The results of the Geographical Mapping experiments. Music from 33 different styles was presented to two instantiations of Maestro, one trained in Chinese music, the other in German. The first and second columns show the style and the number of sample pieces used. The next three columns show the corresponding prediction entropy values for each listener, as well as the difference between the two listeners. Multiple styles from different countries are shown in grey, with the weighted average total for the country shown in black.

results for a given listener.

Looking at the rest of the data, two overall trends emerge. First, each style has its own intrinsic entropy. For example, Japanese and Swiss music seem intrinsically difficult for both the Chinese (5.54 / 4.32) and German (4.71 / 3.67) listeners, while Brazilian and Turkish music are relatively easy to predict (2.91 / 3.14 and 3.08 / 3.03 respectively). Therefore, to compare the two listeners' performance across the various musical styles, the *difference* in prediction performance between the two listeners is calculated and reported in the final column of Table 10.2.

A map can be drawn to indicate to what extent one listener out-performs the other. Such a map is shown in Figure 10.9. The lighter shading indicates countries where the Chinese-trained system performed better, while the darker shading indicates where the German-trained system performed better. Countries shown in green are those for which no data was available.

In measuring the difference between the two systems' performance, a second trend becomes noticeable. The German-trained system out-performs the Chinese-trained system in almost all cases. This is likely due to the fact that most of the music available in the Essen Folk Song Collection is from Western countries, which share a distinctly closer musical heritage with German music than with Chinese music.

As much of the data comes from the European continent, a close-up map of Europe is also drawn for clarity (Figure 10.10). Since the German system out-performs the Chinese system in all of the European countries, the shading is drawn slightly differently, here indicating to what extent the German system out-performs the Chinese system.

Baroni *et al.* state that a correct description of European melody must take into account the presence of different repertoires [6]. The results of the Geographical Mapping experiments show that although most of the European prediction results are within the same range, there is still significant inter-European variability between the music from the different European countries. For example, the listener with a German musical training predicts Italian music (4.29) significantly worse than Dutch Music (3.2).

This methodology also allows for a geographical analysis to take place. For example, on the world map India is close to China both geographically and musically, and Europe as a whole is both musically and geographically closer to Germany. Furthermore, the European map shows that the countries geographically further from Germany also tend to be generally more distant musically.

There exist exceptions, and more experiments should be performed with additional samples of music from the various styles. Still, it is proposed

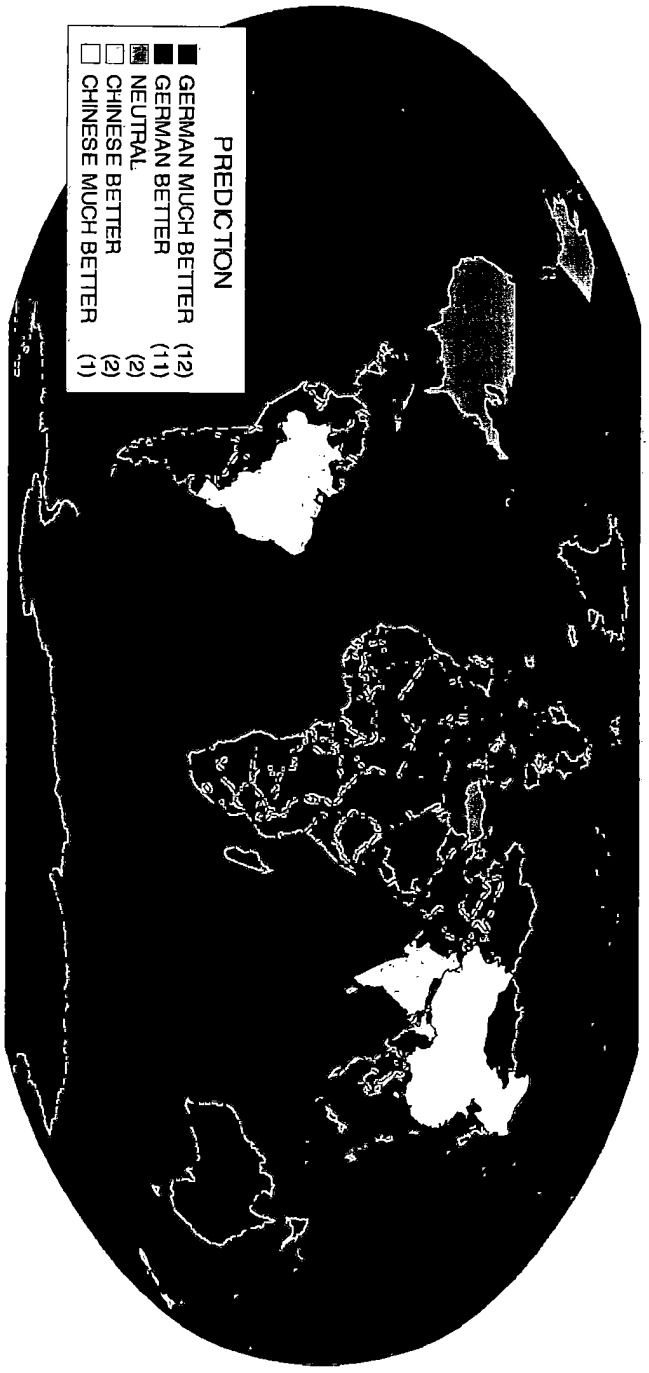


Figure 10.9: Relative prediction performance of Chinese-trained and German-trained systems in predicting music from different parts of the world.

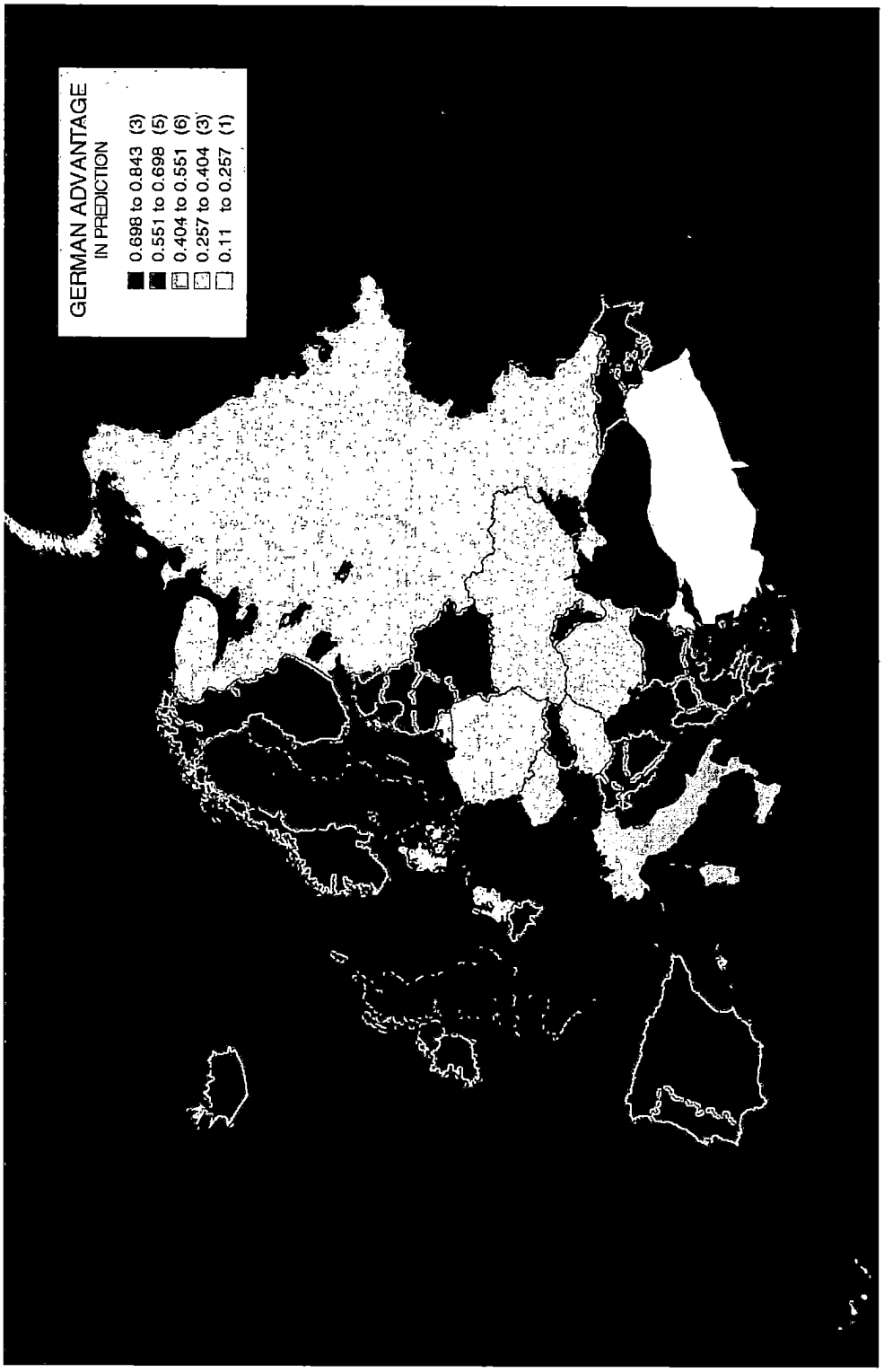


Figure 10.10: A close-up of the map of Europe showing to what extent the German-trained system out-performs the Chinese-trained system.

that this general methodology is useful for finding correlations between geographical, cultural and musical distances.

10.3 Summary

This chapter describes experiments performed with music from multiple styles. The Style Switching experiments show how previous musical experience affects the prediction performance when dealing with music from non-native style. The results also show that with extended exposure, a non-native listener gradually adapts to the new style. A Comparative Listening analysis reveals how native listeners have an advantage over foreign listeners when listening to the same piece.

The Geographical Mapping experiments study how machine listeners from two different musical backgrounds fare in predicting music from 33 different styles. The results show that, in general, native listeners perform better than foreign listeners when listening to native music. Additionally, styles originating from the same country lead to similar performance levels. Finally, the geographical distance of musical origin is generally correlated with differences in levels of prediction performance.

Chapter 11

Related Work

This chapter presents work related to the present research. Theories of music listening and learning are discussed with a focus on a cognitive approach. Then, various machine models of musical learning and musical ambiguity are discussed in relation to Maestro. Finally, multi-agent-based models of music cognition are also presented.

11.1 Theoretical Models Of Music

Although much has been written on music cognition [13, 39, 44, 47, 58, 67, 112], there is to date, no single universally accepted theory. This is due in part to the complexity of the topic, the vast diversity of musical styles, and the dearth of knowledge about cognition in general.

11.1.1 Classical Music Theory

For most of the past few centuries, the study of music theory has focused on developing systems of rules to describe trends and norms occurring in specific styles of music. For example, Aldwell and Shachter's *Harmony and Voice Leading* [2] presents a detailed account of rules governing composition in the classical Western style.

Formal description of musical style is extremely useful for the pedagogical purposes of transmitting the specifics of a musical style to the next generation of musicians. It is also useful for communicating ideas about music from a specific style between two musicians. (See the discussion in Cook [28, p. 3]). However, the limitation of this general approach is that it ignores, according to Lerdahl and Jackendoff [72, p. 2], the obvious fact that music is a product of human activity. Camurri *et al.* similarly remark that several fundamental aspects of common-sense music understanding have their roots in the structure and behaviour of the human auditory system, and not in

the axioms or textbooks on music theory [21]. Additionally, Blacking writes about music as a human capability and invokes Clifford Geertz's statement that "art and the equipment to grasp it are made in the same shop" [18, p. 225].

Maestro approaches music modelling from a cognitive perspective. Two major theories of music that share this cognitive view are now presented.

11.1.2 Grammar: Lerdahl and Jackendoff

The *Generative Theory of Tonal Music* (GTTM) [72] was proposed by Lerdahl and Jackendoff in 1983. The theory is elaborate and complex, and only an outline of the relevant features is presented here. Lerdahl and Jackendoff view the process of listening to music as one of generating appropriate structures from the musical surface (the audio signal). To this end, their theory consists of a musical grammar [5] containing pre-defined rules which are used to analyse music by organising it into mostly-rigid hierarchical structures.

The grammar consists of *well-formedness rules* that dictate the structures that are allowed, and *preference rules* that indicate which of the allowed structures is the most desirable. Although Lerdahl and Jackendoff state in some places which preference rules take precedence over others, in many cases this matter is left undecided, or up to human intuition. There are also *transformational rules*, which apply distortions to the otherwise strictly hierarchical structures.

A sample analysis is shown in Figure 11.1. In the process described by Lerdahl and Jackendoff, the musical data is first grouped and metrically analysed. *Grouping structure* refers to the hierarchical segmentation into motifs, phrases and sections, while *metrical structure* refers to the regular alternation between strong and weak beats at different hierarchical levels. Jones *et al.* [63] have developed GTSIM, a system which simulates the metric and grouping stages of GTTM.

Two reduction trees are then generated. The *Time Span Reduction* indicates the relative structural significance of elements in the piece. This takes into account non-hierarchical elements, such as timbre, dynamics, and motivic-thematic processes, but does not formalise them. The second type of tree is the *Prolongational Reduction* which tracks nested progressions of tension and release. Prolongational reduction takes into account the structural significance of events as presented by the time-span reduction.

Grouping is addressed by Maestro, but meter is not. Tension and relaxation are monitored in Maestro by tracking entropy and agent activation, as described in Section 6.2.3. Since Maestro does not maintain an explicit internal sense of tonality, it cannot generate a prolongational reduction since,

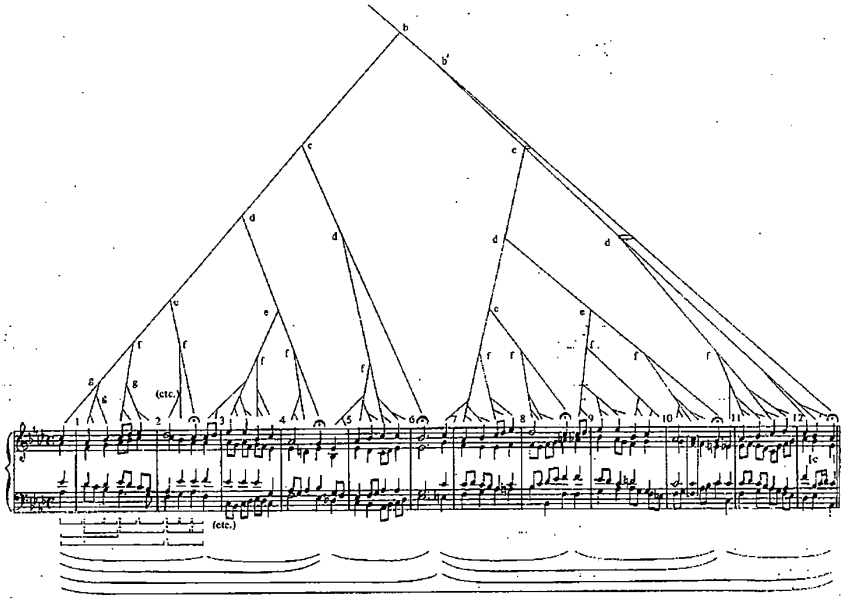


Figure 11.1: A sample analysis using the Generative Theory of Tonal Music. From (Lerdahl and Jackendoff, [72]).

according to Jackendoff [61], a prerequisite for deriving a reductional structure in real time is deriving a sense of tonality. Similarly, since Maestro does not track metric information, it does not generate a time-span reduction.

The benefit of Lerdahl and Jackendoff's rule-based theory is that, like other symbolic models of music cognition, it offers a clear representation of the data structure. This is particularly well suited for the highly structured music in the Western style, for which the rule set was developed. While there is evidence for hierarchical structure in music [41, 113], significant criticism has been levelled against GTTM for enforcing its uniform hierarchical structures on large pieces of music, where it is far from clear that they exist in the music, let alone in listener's minds [81, 117]. Jackendoff comes to terms with some of this criticism in a 1991 paper [61]. Lerdahl and Jackendoff state that their theory is aimed at examining the hierarchical aspects of music. Western tonal music has very rich hierarchies, and is thus very well suited for their approach. Other styles do not have as much inherent hierarchical organisation, and may well be more amenable to different types of analyses.

Lerdahl and Jackendoff state that many of their rules are innate and thus universal, and could in theory apply to all musical styles. However, sometimes the stylistic norms of the idiom simply do not give the rule op-

portunities to apply [72, p. 279]. Despite this qualification, many of the rules in GTTM are definitely idiom-specific. As Lerdahl and Jackendoff state clearly:

> In what follows, we can take as given the classical Western tonal pitch system – the major-minor scale system, the traditional classifications of consonance and dissonance, the triadic harmonic system with its roots and inversions, the circle of fifths system and the principles of good voice leading. Though all of these principles could and should be formalised, they are largely idiom-specific, and are well understood informally within the traditional disciplines of harmony and counterpoint. Nothing will be lost if we conveniently consider them to be an input to the theory of reductions [72, p. 117].

Lerdahl and Jackendoff state that they focus their work on one particular style, with the hope that it will raise ideas about a pan-stylistic theory. Maestro, on the other hand, begins with no style-specific information encoded into the system, giving it the flexibility necessary for handling music from different styles.

In GTTM, Lerdahl and Jackendoff ignore the process of learning. Instead, they take the goal of a theory of music to be a formal description of the musical intuitions of a listener who is experienced in a musical idiom [72, p. 1]. As the goal of the present research is to study music learning, Maestro focuses on the learning process, and begins by modelling an inexperienced listener.

Lerdahl and Jackendoff envision the process of learning as that of developing a grammar [72, p. 296]. Maestro's learning follows this general approach, as its context model is a form of probabilistic grammar, as discussed in Chapter 4.

Finally, GTTM, instead of describing the listener's real-time mental processes, is concerned only with the final state of his understanding [72, p. 4]. This was done to simplify the original problem, and the issue of the processes involved was addressed a few years later by Jackendoff in [61], as described in Section 11.3.2 below. Maestro models the on-line processes involved in music listening, and this approach takes a central role in Maestro's design.

11.1.3 Bottom-Up: Narmour

Another widely cited theory in the field of music cognition is Eugene Narmour's *Implication-Realisation Model*, put forth in [86] and expanded in [87]. In contrast to Lerdahl's and Jackendoff's top-down approach of assigning a

hierarchical reduction tree to a piece of music, Narmour adheres to a more bottom-up approach.

Narmour's is a very complex melodically-based theory, a full description of which is beyond the scope of this discussion. In brief, Narmour theorises that a set of local melodic relations exist between nearby tones in a piece of music. Notes in a melodic line that form a relation can be grouped into a segment. This segment can then be reduced to only its beginning and ending "structural" notes. Narmour states that this reduction mechanism can be re-applied to the structural notes themselves, thus leading to a hierarchical representation of the music. The same relations used to analyse the first level of the musical surface can be used in turn to analyse the reduced version, one level up in the hierarchy.

Narmour's attribution of greater importance to the lower levels of a melodic hierarchy – consistent with his generally bottom-up view – lies in contrast to Lerdahl and Jackendoff's emphasis on the importance of information which appears in higher levels of the hierarchy. Additionally, Narmour describes a looser form of hierarchy, which he calls a *tangled hierarchy*, in contrast to Lerdahl and Jackendoff's more rigid, strongly reduced tree structures.

Narmour describes a top-down style system which consists of learned, replicated complexes of syntactic relations regarding such things as when certain tones or scale-steps are allowed. This operates in conjunction with the bottom-up automatic processes which are innate to all listeners. Narmour discusses the interaction of top-down schematic information with bottom-up musical surface analysis. He states that while the bottom-up analysis is always operating, it can at some points activate a top-down schema. This happens when the music closely enough matches the details of the schema representation. In such an instance, the music can be viewed top-down as well, and expectations can be generated according to the specific rules of the schema. This activation of top-down schemata leading to the generation of expectations is very similar to the approach taken to prediction and parsing in Maestro, as described in Chapters 5 and 6.

Narmour claims that his approach to bottom-up processing is general enough to be applied to music from all styles, and he presents sample analyses of music from different styles. Maestro is similar in its pan-stylistic aims.

Narmour's theoretical description of people's innate musical preferences have been empirically tested with some favourable results [68]. Interestingly, Schellenberg found that the implication-realisation model can be simplified somewhat without losing its predictive capabilities [109].

Narmour's work is impressive in its wide scope and the number of features of bottom-up music cognition for which it can account. However,

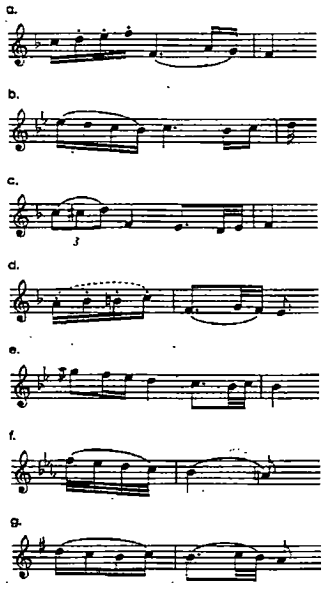


Figure 11.2: Variations of one of Mozart's signatures, identified by David Cope's EMI system. From (Cope, [34]).

his theory does not address the learning processes involved in deriving the top-down style-specific schemas. Maestro's design focuses on handling these learned components of music intuition.

11.2 Machine Models Of Music Learning

The present research focuses on studying music learning. In recent years, machine models of music cognition have been developed to address the issue of music learning from a number of perspectives. A representative sample is presented here, including statistical, rule-based, connectionist and context-model-based approaches.

11.2.1 Statistical: Cope

In his extended programme of research, *Experiments in Musical Intelligence* (EMI) [30, 31, 32, 33, 34], Cope focuses on off-line style induction from musical examples, with the goal of automatic composition in that style. He describes a statistical analysis method for finding a composer's "signatures" – the set of characteristic patterns in the musical surface that can be extracted from large samples of the composer's work (Figure 11.2).

A pattern is considered part of a composer's signature if its degree of repetition is above a certain threshold. The data to be searched is represented as strings of intervals, allowing for transposed repetitions to be noticed. Cope's off-line strategy lies in contrast to Maestro's more cognitively realistic on-line approach.

In searching for patterns, Cope's system makes no use of perceptual cues, but instead segments the input at fixed regular intervals. This fixed-length, non-overlapping approach may miss many patterns, and Cope says that discovery of the right value for this fixed length is difficult [34, p. 38]. Maestro addresses this issue by using adaptive pattern lengths based on perceptual cues in the music, as described in Chapter 3.

Cope's pattern matching techniques include capabilities for handling noise and variability in the data. He defines *range tolerance* as the ability to match intervals to within a certain number of semi-tones, and *error tolerance* as the ability to handle a certain number of out-of-range points before a match fails [32]. These parameters, or *controllers*, are set manually for each run of the system. As described in Section 1.4.6, Maestro performs only exact pattern matching.

Style dictionaries containing the characteristic signatures are assembled by EMI. Cope [31] then describes a top-down composition method based on *Augmented Transition Networks* (ATN) that contain functional descriptions of different musical sections. In general, this involves first laying out a high-level plan for the musical flow of the piece, and then filling it in with the appropriate signatures drawn from the style dictionary.

The performance of Cope's system is judged using the generative approach – namely, to what degree human listeners judge the newly composed piece as sounding like the original style. In this regard, Cope has achieved impressive results. In line with this approach, Cope comments that in music there is no right or wrong, rather, some music just sounds better [34, p. 17]. From the perspective of the present research, however, this generative method of evaluation is, at best, somewhat subjective. Maestro therefore uses the more objective measure of prediction performance to evaluate and track music learning.

Cope points out that his system is able to imitate a style more convincingly given a larger number of examples [33, p. 405]. Maestro's use of large data-sets for learning is in line with this observation.

Due to its purely statistical learning approach, Cope's system is able to handle music from different styles. He claims that the results obtained from EMI suggest that one way of defining musical style is through pattern recognition, and that musical style can be imitated if one can find what constitutes musical patterns [31, p. xii]. This claim is empirically tested and confirmed by Westhead and Smaill [125], who use statistical methods to

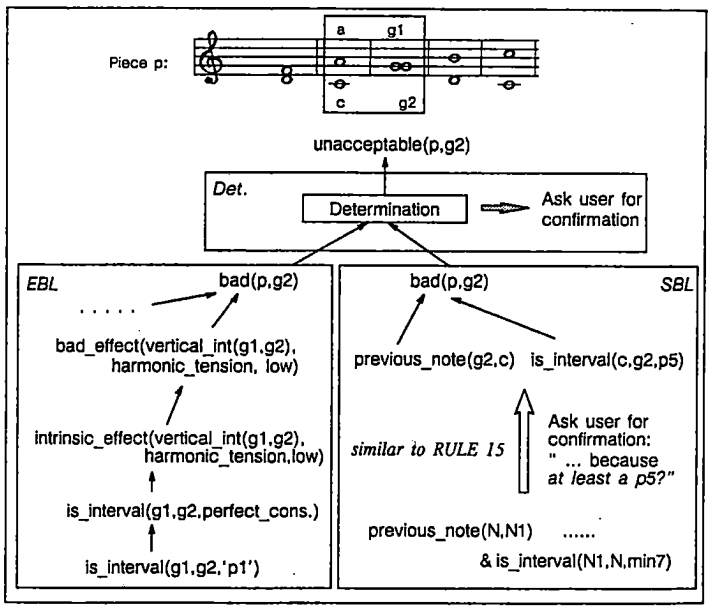


Figure 11.3: A training instance and its explanation in Widmer's symbolic music learning system. From (Widmer, [126]).

successfully learn to discriminate between different styles of classical Western music. They express the need for studying whether or not this claim also holds true for other musical styles. The present research partially addresses this need by performing experiments with music from many different styles.

Much human input is needed in the composing of new work using Cope's system. EMI was developed as a tool to assist a human composer, and still relies on human input and tweaking to achieve good results. In contrast, Maestro performs all the learning on its own, based on its musical experiences.

11.2.2 Rule-Based: Widmer

Many researchers [29, 45, 57, 90] have studied the use of rule-based systems for modelling music understanding. In [127, 126], Widmer notes that "intelligent" learning requires a considerable amount of domain specific knowledge. He proposes that human music cognition and learning rely on a knowledge-base shared by a group of listeners called *musical common sense.*

This knowledge-base includes specific information on such things as notes, keys and scale structures, and can serve as a foundation to be built upon in gaining further musical knowledge. Widmer's system stores the information in a hierarchical knowledge base that contains representations of different layers of musical structure. As a sample task, Widmer has his system learn the rules and constraints of two-voiced counterpoint composition (Figure

11.3). Widmer's more recent work [128, 129] focuses on learning rules for expressive musical performance.

Widmer describes three methods of learning which his system applies to the existing knowledge-base and the incoming musical data in order to generate new knowledge: Deductive rules, Determinations, and Plausibility Heuristics. The first method uses Explanation Based Learning (EBL) to extract general rules from examples, given a set of initial rules. Determinations are statements of general dependencies between attributes which can be used to generate possibilities in the search for new rules. The general kinds of determination to be made are established beforehand. Finally, plausibility heuristics are used to encode general, non-specific knowledge about the musical style, which is helpful in narrowing down the search space for rules.

Widmer's system does not model the real-time dimensions of listening [127, p. 65], but instead focuses on a a kind of *a posteriori* understanding of a piece after it has been heard [127, p. 52]. In contrast, one of Maestro's principle aims is to model the real-time processes of music listening.

The clear knowledge representations and well-defined processing methods inherent to symbolic systems such as Widmer's make their operation easy to follow. A symbolic system can be endowed with a rule-base that allows it to perform complex, hierarchical analysis on musical data. For the purposes of musical learning, however, these rules can also have the undesirable effect of limiting systems to the specific style they were designed to analyse, and this makes the systems less flexible toward processing music from different styles. The design of such rule-based systems requires a very detailed understanding of the environment in which the intended learning is to take place. The inherent uncertainty and variability of the learning environment that is music, and the lack of a unified pan-stylistic theory of music cognition, together demand a flexibility which the symbolic approach cannot easily provide.

With respect to the present work, it is important to note that while Widmer's model does show evidence of learning, it starts out with the fundamental information about a style already present in the form of the knowledge base. In contrast, the present research focuses on learning to listen to music from a certain style without possessing this knowledge ahead of time, but learning it from experience instead.

Widmer states that his work can also be useful in testing theories of music by encoding them into rules and seeing how well they perform. An analogous approach is taken in the present research; Maestro is an attempt to see how well certain theories of music learning perform when implemented in a machine model.

11.2.3 Connectionist Approaches

At the other end of the machine learning spectrum are connectionist systems, which have been widely used in recent years for modelling music cognition. Artificial neural networks have been used in efforts to learn systems of tonality by deriving mappings between pitches and chords or keys [10, 51]. Gang and Berger use a connectionist model to study music expectation [49]. A good review of these and other uses of neural networks is found in Leman [71], as well as the very recent collection by Griffith and Todd [52].

Connectionist systems boast good performance in their various learning tasks, and it can be argued that these systems are in fact successfully picking up elements of the style information from their training data. Another advantage of connectionist systems is the somewhat improved flexibility for dealing with music from different styles.

However, the general drawback of the connectionist approach is that it is often difficult to understand how or what the machine is learning. A goal of this research is to study how the system is going about learning the specific characteristics of a musical style. This type of analysis is made difficult by the holistic approach of connectionist systems. Another drawback of the connectionist approach, according to Jackendoff, is that complex structures are more difficult to represent [61, p. 201]. Maestro's (non-connectionist) design allows it to learn in many environments, while maintaining clarity as to what and how it is learning at each stage.

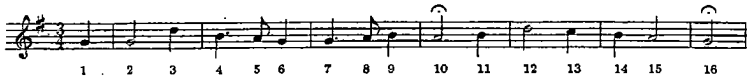
11.2.4 Context Models: Conklin and Witten

In their seminal program of research into music prediction stretching over a few years [26, 27, 77, 131, 132], Conklin and Witten set out to develop a machine model that learns to predict melodic pitches based on past listening experiences. The system achieves impressive results in predicting Bach Chorale melodies, performing almost as well as human subjects given the same task, and displaying a high correlation with human performance.

Conklin and Witten's system stores segments of pitch information into a context model (Chapter 4) and the performance is evaluated using an information theoretic framework based on calculating a type of entropy (Chapter 5).

In [132], Conklin and Witten compare the music prediction capabilities of humans and computers. They perform their tests only on one-step-ahead predictions. Due to its variable order context model, described in Chapter 4, Maestro is able to generate appropriate multiple step predictions.

The input to Conklin and Witten's system is pre-segmented with fermatas, and they state that this provides very strong clues about the properties of the next event. They assume that the beginning of a phrase is the



Type	Event number															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
st	8	12	20	24	30	32	36	42	44	48	56	60	68	72	76	84
pitch	67	67	74	71	69	67	67	69	71	69	71	74	72	71	69	67
duration	4	8	4	6	2	4	6	2	4	8	4	8	4	4	8	8
keysig	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
timesig	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
fermata	F	F	F	F	F	F	F	F	F	T	F	F	F	F	F	T
deltast	⊥	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
gis221	⊥	4	8	4	6	2	4	6	2	4	8	4	8	4	4	8
posinbar	8	0	8	0	6	8	0	6	8	0	8	0	8	0	4	0
fib	F	T	F	T	F	F	T	F	F	T	F	T	F	T	F	T
seqint	⊥	0	7	−3	−2	−2	0	2	2	−2	2	3	−2	−1	−2	−2
contour	⊥	0	1	−1	−1	−1	0	1	1	−1	1	1	−1	−1	−1	−1
referent	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
intfref	0	0	7	4	2	0	0	2	4	2	4	7	5	4	2	0
inscale	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
intfib	0	0	7	0	−2	−4	0	2	4	0	2	0	−2	0	−2	0
intfip	0	0	7	4	2	0	0	2	4	2	4	7	5	4	2	0
intphbeg	0	0	7	4	2	0	0	2	4	2	0	3	1	0	−2	−4
thrbar	⊥	0_4	⊥	4_{12}	⊥	⊥	-4_{12}	⊥	⊥	2_{12}	⊥	5_{12}	⊥	-3_{12}	⊥	-4_{12}
lphrase	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	40	⊥	⊥	⊥	⊥	⊥	28
thrph	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	4_{48}	⊥	⊥	⊥	⊥	⊥
thrqu	⊥	0_4	7_8	-3_4	⊥	-4_8	0_4	⊥	4_8	-2_4	2_8	3_4	-2_8	-1_4	-2_4	-2_8

Figure 11.4: The multiple viewpoints used in Conklin and Witten's context modelling approach. From (Conklin and Witten, [26]).

event immediately following an event under a fermata[1] [27, p. 62]. On the other hand, segmentation in Maestro, as with humans, is performed by the system on-line according to perceptual cues.

Conklin and Witten do not only store pitch in their context model. Instead, they adopt a multiple-viewpoint approach [27] in which the data is stored in a number of different representations (Figure 11.4). The different viewpoints include pitch, time signature, key signature, fermatas, start time, and duration. According to Conklin and Witten [131], it is logical to expect predictions to improve when various sources of knowledge about an event stream are correlated and used together.

The problem with this approach from the perspective of music learning is that some of these viewpoints rely on information not readily available in the musical surface. For example, human listeners derive the time and key signatures on-line from the time and pitch information, but Conklin and Witten pre-annotate the input with this information. This goes against the realistic input specification of Maestro's design, described in Section 2.2.1.

[1] Apart from this pre-segmented phrasing used for deriving some viewpoints, Conklin and Witten's system uses a brute-force segmentation strategy for storing data in the context model, as described in Chapter 3.

Conklin and Witten also describe a set of *derived types*, which are additional viewpoints created from the basic ones listed above. There are too many to discuss here, but many rely on style specific knowledge for their derivation. As Conklin and Witten explicitly state, the central idea behind viewpoints is to use background domain knowledge to derive new ways of expressing events in a sequence [27].

The rhythm, key, time and fermata information for the current event, in addition to all previously predicted complete events, is available to the system before predicting a certain pitch (Darrell Conklin, personal communication). While Maestro also predicts only pitch, it does not have access to any auxiliary information ahead of time.

Conklin and Witten actually view the fact that they do not incorporate further background information into their system a "shortcoming" [132, p. 78] – since their aim is optimal prediction performance from a machine model (see [131, p. 57] and [132, p. 79]), they welcome any extra knowledge that aids prediction. Conversely, the goal of the present research is the study of a more cognitively realistic system for music learning. Since this *a priori* knowledge is in fact the subject of the learning, it is not appropriate to include it in a learning system.

Due to the style-specific knowledge encoded in it, Conklin and Witten's system lacks the flexibility to handle music from many different styles. While in an earlier paper they state that their goal is to devise a methodology capable of handling different genres, [131, p. 62], in their last paper they point out that this goal has not yet been reached and that in order to be significant as a general-purpose machine learning tool for music, the system should be applied to musical domains wider and more adventurous than the Bach Chorale melodies [27, p. 71]. The Bach Chorales are chosen in part due to their general display of good melodic form [27], and this makes the task and study of music style induction easier. However, people are capable of learning different styles, and in keeping with this, experiments with Maestro are performed using music from many different styles.

Finally, Conklin and Witten remark that system performance improves as more training Chorales are seen [132]. Thus, the 100 Chorales on which their system is trained are not sufficient for studying the complete process of music learning. As the aim of Maestro is to study music learning, much larger data-sets containing thousands of pieces are used to ensure that the system undergoes a more complete process of learning.

11.3 Machine Models of Musical Ambiguity

Ambiguity plays a central role in music listening, especially in the context of learning. Two representative approaches to handling ambiguity in music are now presented.

11.3.1 Tanguiane

Tanguiane's work on *artificial perception* [118] focuses on real-time automated notation of performed music. His system deals with segregating different instruments from a sound stream as well as tracking rhythms and tempos.

Tanguiane's work is based on two principles: grouping and simplicity. The *grouping principle* states that similar configurations of stimuli in the data can be located and used to form high level configurations. Gestalt principles such as parallel motion and continuity are used to organise the data. From these Gestalt principles, grouping ambiguity often arises, suggesting a number of possible representations of the data. The *simplicity principle* states that the ideal representation of the music is the one that is least complex according to the Koglomorov minimum representation – the one that uses the least memory. Tanguiane thus formulates the problem of musical perception in terms of optimal data representation. Data is stored as generative sound elements and their transformations. Guided by the simplicity principle, these representations are built up into hierarchies. Tanguiane claims that these hierarchies are a first step to musical understanding.

The driving force in Tanguiane's system is finding optimal representations. In contrast, Maestro determines the structure of the music based on finding repeating patterns in the input, as described in the discussion of Maestro's parsing stage in Chapter 6.

Tanguiane's system has no long-term learning, which is Maestro's primary focus. Tanguiane's methods are applied to low-level audio signal transduction for automatic notation of music. They are not necessarily suited for the task of higher level music cognition considered here.

Most importantly, in handling ambiguity, Tanguiane's system immediately resolves it, while Maestro maintains multiple hypotheses of interpretation. The motivation for Maestro's approach to ambiguity comes primarily from the work of Jackendoff, described next.

11.3.2 Jackendoff

Following up on the original GTTM work, Jackendoff published a paper in 1991 entitled *Musical Parsing and Musical Affect* [61], pointing out some limitations of the original work. In it, he states that GTTM is intended as an account of the experienced listener's final-state understanding of a piece – the structures that the listener can attain, given full familiarity with the piece and with the style, and no limitations of short-term memory or attention [61, p. 200].

Jackendoff emphasises that listening is an on-line process aimed at de-

riving appropriate structure from the musical surface through a process he calls "musical parsing". He points out, however, that GTTM only reports the final state of this structure and in no way addresses how it is derived over the course of listening to the piece. Jackendoff thus arrives at the need for a "processing model" able to show how the principles of the listener's internalised musical grammar can be deployed in real time to build musical representations.

Jackendoff discusses the issues faced by such a model, focusing especially on on-line parsing and ambiguity handling. He says that one of the fundamental problems facing the processor is the indeterminacy of the analysis at many points – an indeterminacy that sometimes cannot be resolved until considerably later in the music [61, p. 210].

After considering three different types of parsers to deal with this ambiguity (see Section 6.1.3), Jackendoff chooses the *parallel multiple analysis model*: when handling ambiguity, processing splits into simultaneous branches, each of which computes an analysis for one of the possibilities. When the plausibility of a particular branch drops below a certain threshold, it is abandoned. The branch remaining at the end represents the best analysis of the music.

Jackendoff also deals with the issue of cognitive load, and how many interpretations can be maintained by the listener at once. To choose between different hypotheses, a selection function is employed, based on frequency, plausibility or structural simplicity. These issues and others are discussed more fully in the discussion of parsing in Chapter 6.

In dealing with ambiguity, Jackendoff's general approach of maintaining multiple hypotheses lies in contrast to Tanguiane's immediate-resolution method. In keeping with its cognitive realism design principle, Maestro bases its strategy of handling ambiguity on Jackendoff's approach.

Both the original GTTM work and Jackendoff's 1991 work have the system start out with an initial set of knowledge in the form of a rule base. Maestro departs from Jackendoff's approach here in that it aims to learn this information from experience.

11.4 Agent-Based Models

This research draws on and extends the recent work in agent-based music cognition systems [55, 82, 102, 103]. As the term *agent* is used in a wide variety of contexts [40], it is appropriate to begin this discussion with some definitions:

Agent An autonomous entity able to take certain actions to accomplish a set of goals.

Multi-Agent System A collection of independent agents, each working towards its own goals. The agents in such a system might operate alone, interact with each other, cooperate, compete, and can also learn. From the actions and interactions of the individual agents, the complex behaviour of the system as a whole emerges. (See [40]).

The concept of societies of agents arose from the work of Minsky and Papert. In *The Society of Mind* [83], Minsky proposes a theoretical framework of many interacting agents, each performing different tasks. A group of agents performing related tasks can come together to form an *agency.*

A number of the intrinsic properties of multi-agent systems make them highly suitable for the study of music cognition and learning [82, 103], namely: competition, cooperation, and emergence. Agents with different views of the music *compete* with each other to make themselves heard. Agents can also *cooperate* to help each other to achieve common goals. Out of the interactions of the various agents, the desired system-wide behaviour *emerges.*

Within agent-based models of music cognition, an important distinction can be made between processing agents and representative agents. In the first approach, the various *processes* of music cognition are divided up among different agents. This approach describes the agents in Rowe's system [103] as well as the Knowledge Sources used by Jones *et al.* in [63]. In the second approach, individual agents not only process music, but actually *represent* different interpretations of the music and compete with each other to have their view chosen as that of the system. Maestro is designed according to this latter approach.

Agents are not just passive data structures. Their ability to perform actions to achieve their goals makes them well suited for implementing the active listening approach introduced in Section 2.2.7. As Rosenthal states, memory and process are contained in one structure [98, p. 318].

Learning within multi-agent systems is also an active area of research [12, 40, 124]. Inter-agent and intra-agent learning techniques can be used to improve the system's performance with experience.

As the multi-agent system paradigm is a broad, general approach, it is sometimes difficult to formalise its contribution to a system's design. Rosenthal explains:

> The idea of a society of agents has strong implications for the structure of our model. First of all, rather than thinking of the model as a single, integrated process that produces, say, encoded descriptions of rhythms from raw data, we may think of the model as a collection of independent, specialised processes, each of which attacks the part of the incoming data that corresponds to its speciality. Although this is perhaps more a difference in how the workings of the model are explained than a difference

> in the workings themselves (because most computer programs may be thought of as being composed of smaller pieces), the distinction is nevertheless a valuable one and has a profound effect on how one thinks about and proceeds to build such a model [98, p. 317].

Moreover, the incorporation of the multi-agent system paradigm into Maestro's design provides for the explicit maintenance of multiple simultaneous interpretation hypotheses when handling ambiguity. In keeping with Jackendoff's approach, Maestro also contains a distributed agent-based parsing algorithm that maintains multiple simultaneous hypotheses of interpretation.

11.4.1 Minsky

In *Music, Mind and Meaning* [82], Minsky proposes a theoretical model of music cognition, focusing on a procedural description of music. He argues that the attempts to find a unified theory of music accounting for all musical styles have failed because music universals cannot be found on the musical surface level. Rather, the universals are found in the processes that generate and perceive the musical surface.

In keeping with this approach, Minsky proposes a model in which different agents process the music in parallel, building up multiple representations of the music in the system. Agents represent different views of various patterns occurring in the music, and so similar patterns in the music activate similar agents.

According to Minsky, learning occurs through the addition of new agents. Some agents can serve to connect other agents, and agents can cooperate. For example, one agent's activation can reduce the threshold of activation for another agent, whose pattern is expected to occur in succession.

Minsky's is only a theoretical model, and is not implemented into a system. Maestro is designed to test certain key aspects of Minsky's theory by implementing them into a working system and performing experiments with real musical data.

Minsky also proposes some experiments that involve raising simulated infants in traditional musical cultures. From here stems the motivation for some of the multi-style experiments performed in Chapter 10.

11.4.2 Rosenthal

In extending Minsky's work, Rosenthal describes an agent-based system for modelling and structuring rhythmic information in music [98]. The system records previous rhythmic patterns it has seen, and creates agents called

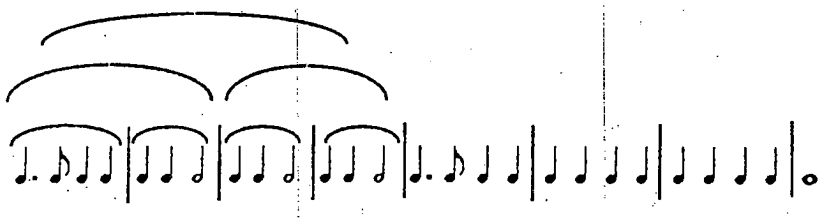


Figure 11.5: Rhythm recogniser agents making sense of a rhythmic selection in Rosenthal's system. From (Rosenthal, [98]).

recognisers to track these patterns when they reappear. Since memory and process are contained in one structure, the hierarchically organised recognisers constitute a structural description of the rhythm in the piece (Figure 11.5).

Rosenthal uses the GTTM grouping rules to segment the rhythmic input in order to determine when recognisers should be created. Even though Rosenthal states that humans make use of pitch information to aid in rhythmic processing, pitch information is entirely ignored in Rosenthal's system. Similarly, Maestro predicts pitch, but does not store timing information. However, Maestro does make use of timing information for the purposes of segmenting the pitch data for storage.

In order to organise the recognisers into an appropriate structure, Rosenthal's system relies on the *law of return* – two occurrences of the same pattern, interrupted by other rhythmic material. This configuration is said to form a structural unit which is then able to be organised hierarchically. Rosenthal makes an interesting observation with regard to this method:

> "Tension" rises in the model when there is a proliferation of recognisers that are not yet organised into a higher level structure. In a human, this corresponds to having more items in short-term memory than one can comfortably handle. [98, p. 327]

This idea is adopted and extended in the present research. Maestro measures ambiguity and tension by monitoring the number of listening agents active and predicting at any one time. This is discussed further in Chapter 6.

In handling ambiguity, Rosenthal's system does not maintain multiple hypotheses. Instead he notes that a limitation of the model is that it only retains one interpretation of what it has heard. However, as a possible improvement to the system, Rosenthal suggests that several possibilities could be stored in the expectation that subsequent events would confirm

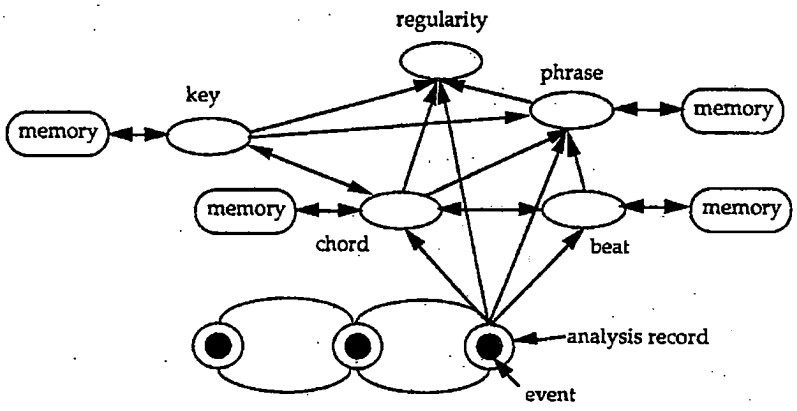


Figure 11.6: Hierarchical music processing agents in the listener component of Rowe's Cypher system. From (Rowe, [103]).

one of them. This is the approach taken by Maestro.

Like Minsky, Rosenthal sees music learning as a process of using existing agents and adding new agents. Rosenthal does not run his model on music from other styles, and therefore he states that it is not clear to what extent the program models listening to music in general and to what extent it is limited to the context of the Western musical tradition. The present research addresses this issue by performing experiments with music from different styles.

11.4.3 Rowe

Rowe's Cypher system [100, 102, 103, 104] was developed as a real-time, interactive agent-based machine system, capable of listening to and accompanying a live musical performance. Cypher consists of two main modules – a listener and a player. The listener processes the incoming musical data and communicates certain information to the player, which then uses this information to control the accompaniment. Cypher also has an internal critic that monitors the system's output using a pre-programmed set of aesthetic preferences.

In Cypher, different agents are used to perform the various tasks involved in interactive music listening and composition. The listener module is hierarchical and has two main levels of listening agents. The first level consists of feature agents that produce *feature streams* containing density, speed, loudness, register, duration, and harmony information from the musical surface. This is similar to the multiple viewpoint approach discussed by Conklin and Witten.

The second level consists of key and chord agents that rely on data from the level-one feature agents in order to determine the appropriate key and chord information. Thus, in Rowe's system agents are arranged into a processing hierarchy, instead of a data-representational hierarchy (Figure 11.6).

The Phrase Agency, a group of related agents, is responsible for segmenting the musical data into phrases for storage into memory. The phrase boundaries are determined according to the discontinuities in the various feature streams. The focus and decay method is used to adaptively adjust to the changing musical dimensions, as described in detail in Chapter 3. The magnitudes of discontinuity from the various feature streams are combined using a fixed weighted average. Thus, Cypher immediately resolves any segmentation ambiguity. A similar method of finding phrase boundaries is used in Maestro's segmentation stage, except that the segmentation cues from the various feature streams are not integrated using a weighted average. Instead, each stream is dealt with independently, and the resulting multiple segmentation hypotheses are all stored by the system. In this way, Maestro handles segmentation ambiguity.

Cypher is also equipped with certain limited pattern processing capabilities [102, Ch. 6]. The listener component maintains a codebook (effectively a context model) of monophonic pitch patterns and harmonic progressions detected in the data and segmented as above. It then employs a noise-tolerant pattern matching technique.

Predictions are made when previously spotted patterns begin to appear again in the input. The listener communicates expectations to the player when it detects patterns recurring. A long-term memory stores patterns from one piece to the next and deletes patterns if they are not used for a long time.

Rowe mentions ([102, p. 163]) that spurious phrase boundaries could interrupt the system from noticing a desired pattern. He also points out that patterns spanning across phrase boundaries simply cannot be found. These issues pose less of a problem in Maestro, where multiple segmentation hypotheses are maintained.

The agents in Rowe's system are designed to function with Western music. For example, the chord and key agents have the Western system of tonality hard-coded in their internal analytic processes. This affects Cypher's ability to deal with music from other styles. Cypher allows for some of the rules controlling composition to be manually tuned by adjusting connections between various listening and composing agents. This learning, however is not driven by the system's experiences, as would be necessary for a self-contained model of music learning.

11.4.4 Hiraga

Hiraga [55, 56] explores the use of representative agents for handling pitch information in melodies. *Coarse receptor agents* are used to identify structure for the purposes of segmenting a melody. The agents monitor a melody entered as streams of pitch and time information. Each agent represents a different primitive relationship between pitches or time spans: same, sequence-up, sequence-down, remote-steps-up, remote-steps-down. Hiraga states that these primitives are congruent with aspects of Narmour's Implication-Realisation model.

Hiraga sets out three design requirements for his system:

- Incremental processing – the system continually reports (partial) results as it processes the music.
- Robustness – the system can handle change, errors and different styles, and is not governed by the presupposition of any musical style.
- Minimal framework – as an initial study, the data is kept simple, limited to only pitch and onset times of notes.

These three factors all play a major role in Maestro's design.

Hiraga's agents are instantiated whenever the relationship they embody accurately describes the current notes. The activation of an agent over a group of notes (e.g., the sequence-up agent over the sequence $ABCDE$) suggests that those notes should be grouped together.

Segmentation is thus performed by agent activation and interaction. If all the active agents agree, the segmentation is clear. If, however, the agents disagree, this leads to ambiguity. In keeping with Meyer's view [115, p. 190], Hiraga comments that segmentation ambiguity enables continuous flow in the music. Ambiguity is resolved using judgements that are dynamic and context dependent, such as the relative metric weight of notes.

Maestro's method of agent activation is similar to Hiraga's in that agents are dynamically generated and invoked by the relation they respond to [55]. Additionally, Hiraga's method of grouping by agent activation is also similar in approach to Maestro's parsing stage.

However, Hiraga uses agents representing *pre-encoded primitive* relations to segment the data, while Maestro, focused primarily on studying music learning, uses agents representing *learned patterns* to parse the data according to the system's previous experiences.

In his later work [56], Hiraga focuses on pattern matching as an essential component of music cognition, and addresses the *circularity problem* that results from the interaction between discontinuity-based segmentation and

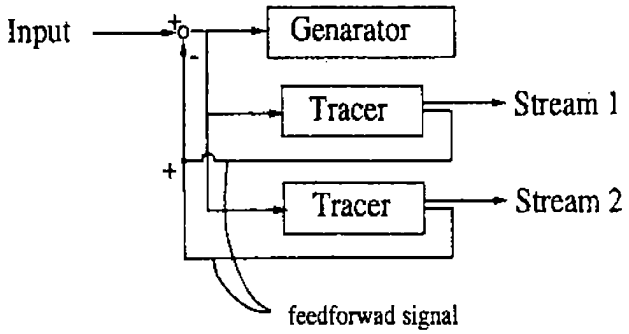


Figure 11.7: Auditory stream tracer agents. From (Nakatani, [85]).

repetition-based segmentation. The circularity problem is formally introduced in Section 3.2.6 and addressed in Section 6.2.4.

11.4.5 Nakatani

Nakatani *et al.* [85] present a system of agents used to track multiple streams in analog audio signals. *Tracer agents* are created by *generator agents* to track multiple audio streams which rise and fall in pitch. A key feature of Nakatani's system is subtractive competition, whereby each tracer agent subtracts the stream it is modelling from the overall signal, thus not allowing other agents to see it, lest they too begin to track it (Figure 11.7).

Separate monitor agents control the agent population. Tracer agents with nothing to track are attenuated, and those modelling the streams of others are terminated. After all the current tracer agents have subtracted their modelled streams from the data, if a significant residue remains, a new tracer agent is launched by a generator agent to track it.

Maestro is closely related to the work of Nakatani. In both systems, agents are used to both process and represent the data. Agents with different goals compete in performing the overall system-wide task. Nakatani's system is focused on stream segregation, while Maestro is focused on prediction and parsing. One possible application of Nakatani's subtractive competition would be to upgrade Maestro to process polyphonic musical data. The stream segregation capabilities would be then be useful for tracking patterns in the different voices.

11.5 Summary

Classical music theory often ignores the essential connection between music and the humans that create and appreciate it. More recent work, such as

that of Narmour and Lerdahl and Jackendoff, approaches music theory from a more cognitive perspective. Maestro follows this cognitive approach.

Machine models of music learning such as Conklin and Witten's have also been designed in accordance with some cognitive principles. However, for the most part these systems are primarily intended for achieving optimal machine performance at the expense of cognitive realism. Maestro is an attempt to develop a model of machine learning that is guided by cognitive principles.

Recent work also directly addresses musical ambiguity. Specifically, Jackendoff proposes that multiple, simultaneous hypotheses are maintained in dealing with ambiguity. Maestro is in part intended as an implementation of certain key aspects of Jackendoff's theory.

The intrinsic properties of multi-agent systems make them highly suitable for modelling music cognition, especially in the context of ambiguity and active listening. Based on the ideas of Minsky, Rosenthal, Rowe and Hiraga have explored ways of using agents to model certain aspects of music cognition and learning. Maestro extends this work, focusing specifically on music learning and cognitive realism.

Chapter 12

Conclusions

Maestro has been developed as a machine model of music cognition and learning. It is designed primarily to enable the performance of music learning experiments that would be virtually impossible to conduct with human subjects. To this end, machine modelling provides the necessary control over prior musical experiences and the ease of continual monitoring that are required for a systematic study of music learning. Maestro also tests certain theories of music cognition and learning by implementing them in a working system and performing experiments with actual musical examples.

Maestro is an attempt to develop a model of machine learning that is guided by cognitive principles, and is not intended for achieving optimal machine performance at the expense of cognitive realism. The experiments described in this dissertation utilise large data sets of music from different styles, thus enabling the study of a more realistic music learning scenario, and allowing multi-style experiments to be carried out. It has been common practice in other programmes of research to incorporate *a priori* style-specific information in order to help achieve optimal performance with a specific musical style. In order to maintain its pan-stylistic capabilities, Maestro does not include any such style-specific information *a priori*.

Ambiguity is an essential aspect of music cognition, especially in the context of learning. Maestro handles three types of ambiguity: segmentation ambiguity, prediction ambiguity and parsing segmentation ambiguity. The intrinsic properties of multi agent systems make them highly suitable for modelling music cognition, especially in the context of ambiguity and active listening. Based on the ideas of Minsky and systems such as Rosenthal's, Rowe's and Hiraga's, Maestro extends the research into ways of using agents to model certain aspects of music cognition and learning, focusing specifically on music learning and cognitive realism.

12.1 Contributions

The following are the main technical contributions of the research presented in this dissertation:

- *Bottom-up Segmentation*: Maestro's segmentation stage is based on tracking three bottom-up perceptual cues, and does not utilise any pre-annotations or *a priori* style-specific information as in previous systems. The focus and decay methodology used by Rowe is adopted for this purpose, and modified to better avoid finding spurious phrase boundaries.

- *Segmentation Ambiguity*: In contrast to previous systems, Maestro stores all possible segmentations in its model. This *partial overlap* strategy is in keeping with Maestro's general approach of maintaining multiple hypotheses when faced with ambiguity.

- *Perceptually Guided Segmentation*: It is hypothesised that certain segmentation points result in models that are more efficient for the purposes of prediction, and further that Maestro's perceptually guided segmentation strategy identifies such points of segmentation. An experimental method, *N-Note Segmentation Shifting*, is developed to compare the efficiency of different segmentation strategies for modelling for prediction. Experiments with short term memory context models are highly consistent with PGS hypotheses, and some of the observations would be difficult to explain otherwise. Experiments with long term memory are still consistent with PGS theory, but the large differences in model sizes prevents a fair comparison of efficiency being made.

- *More Realistic Context Models*: The present research attempts to make the context modelling paradigm used by Conklin and Witten more cognitively realistic. This is accomplished primarily by storing the variable order context segments suggested by Maestro's perceptually guided segmentation. This strategy allows Maestro to generate appropriate multiple-step-ahead predictions and to perform proper repetition-based parsing. Additionally, Maestro's model stays within more realistic size constraints.

- *Activated Modelling*: In keeping with the active listening approach and with Jackendoff's theory of maintaining multiple simultaneous hypotheses when faced with ambiguity, Maestro incorporates the multi-agent system paradigm into its design. The context model is activated by instantiating various segments into autonomous reactive listening agents, which then go on to predict and parse the musical information.

The active modelling approach implements many of Minsky's theories into a working system and tests them with actual musical examples.

- *Multiple-Step-Ahead Prediction*: Previous studies focus only on predictions made one-step ahead of time. Due to the variable order contexts in Maestro's model, the listening agents generate appropriate multiple-step-ahead predictions, in accordance with the current musical context.

- *Extended Entropy-Based Performance Measures*: Conklin and Witten use one type of entropy to measure prediction performance. Two types of entropy are used in Maestro: overall entropy measures the flatness of the generated probability distributions, while prediction entropy reflects the degree of surprise experienced when observing certain events. To allow entropy to be measured, the zero-frequency problem is addressed by blending the system's predictions with a flat probability distribution.

- *Agent-Based Parsing*: Maestro's parsing is performed in a distributed fashion by the various listening agents. It is, in effect, a distributed implementation of breadth first, bottom-up, left-to-right, partial, optimal chart parsing. In keeping with Jackendoff's theory, multiple parsing hypotheses are maintained and reconciled by the agents, who compete and cooperate with one another, each pursuing its own parsing goal. The desired system-wide parsing behaviour emerges from the interactions between the individual agents.

- *Realistic Deterministic Musical Parsing*: Humans use certain selection factors to resolve situations with persistent ambiguity. For resolving parsing ambiguity, Maestro uses preference rules related to right association and lexical preferences, adopted from the field of Natural Language Processing.

- *Retrospective Listening*: Ambiguity often leads to delays in on-line processing, and both Jackendoff and Berent and Perfetti state that this can lead to retrospective listening. Maestro's parsing stage is capable of displaying retrospective listening when faced with persistent ambiguity.

- *Large Data-Sets*: Previous work on music prediction has used corpora containing on the order of 100 songs. Much larger-scale music learning experiments are conducted here with corpora containing thousands of songs taken from the Essen Folk song collection. These constitute an experience base more similar to that of a human listener growing up in a certain culture.

- *Three-Fold Analysis Framework For Music Learning Experiments*: Three methods are developed and used to analyse the results of the music learning experiments: context model growth, the number of predictions generated, and the prediction performance. This framework is also used to analyse predictions of various forecast horizons.
- *Prediction Ambiguity*: Different agents generate various predictions, as suggested by the current musical context. These predictions are integrated into a probability distribution, thus assigning credit for multiple simultaneous predictions. Predictions are weighted by two factors: reliability and certainty.
- *Entropy-based ambiguity classification*: Two types of entropy and a measure of agent activation are used to study ambiguity. A plot comparing overall entropy with agent activation is shown to identify two types of ambiguity: ambivalence and uncertainty.
- *Model Maturity*: A dual-entropy profile is shown to provide a measure of the level of training, or 'maturity' of a model.
- *Circularity Problem*: Segmentation can rely on finding repeating patterns in the data. Maestro deals with the circularity problem inherent to repetition-based segmentation (parsing) by performing a *directed search* for patterns, based on perceptually guided segmentation.

12.2 Lessons Learned

Maestro's capabilities enable large-scale music learning experiments to be performed. The following note-worthy results were obtained:

- *Study of the Learning Process*: Unlike previous research which focused on analysing the final state of a trained music learning system, experiments performed with Maestro study the *learning process* that occurs when listening to music. The results show clear learning curves: with training, the model growth rate decreases, the number of predictions increases and the prediction performance improves.
- *Larger data sets*: The larger data sets used for experimentation enable the study of a more complete music learning process. The results show a learning limit that is approached asymptotically with increased training.
- *Multiple-Step-Ahead Predictions*: The results show that, as expected, prediction performance drops with increasing forecast horizon. For the learning set-up used in these experiments, it appears that predictions up to three or four steps ahead were useful.

- *Style Switching*: The Style Switching experiments show how previous musical experience affects the prediction performance when dealing with music from non-native style. The results also show that with extended exposure to a new style, a non-native listener gradually adopts to the new style.

- *Comparative Listening*: The Comparative Listening experiments study how two listeners with different musical backgrounds perform when listening to the same piece. The results show that the native listener has an advantage over the foreigner, and that the results of the two listeners are correlated due to the intrinsic entropy of the individual songs. As the foreign listener learns the new style, the performance levels of the two listeners gradually converge.

- *Geographical Mapping*: The Geographical Mapping experiments study how machine listeners from two different musical backgrounds fare in predicting music from 33 different styles. The results show that training with related musical styles better prepares a listener for listening to a new style. The results also show that the geographical distance of musical origin is correlated with differences in levels of prediction performance.

12.3 Opportunities for Future Work

Significant contributions have been made over the course of this research, and the potential for further research is promising. Maestro could be extended to perform inexact pattern matching, allowing for a certain level of tolerance when matching the input to previously stored patterns. This would improve the cognitive realism, but would raise new issues such as how dissimilar two patterns must be to justify the instantiation of separate listening agents.

If polyphony is considered, the dimensionality of the problem would be significantly increased. Many interesting issues would arise, such as modelling interactions between the various voices. Explicit models of harmony and tonality can also be added. To maintain Maestro's pan-stylistic potential, these would have to be general enough to learn the regularities of different tonal systems.

Maestro can be developed further to handle hierarchical musical structure. Hierarchical linking agents could represent the concept that one listening agent is typically activated after another. Prediction horizons can thus be extended, and parsing competitions can be modified to prefer hierarchical structures.

The context models can be examined to see if Narmour's innate principles of melodic implication (Registral Direction, Interval Difference, Regis-

tral Return, Proximity and Closure, mentioned in [68]) emerge from listening to music of various styles.

The system could be configured for real-time interaction with a live MIDI device. Maestro is already prepared for such an interaction. As it takes in only the musical surface, no significant pre-processing would need to be done. The adaptive capabilities of focus and decay should prove capable of handling realistic user input. Maestro's segmentation stage could then be upgraded to deal with dynamics information, which was found not to be useful in the experimental data available for this research.

Further experiments could be performed. The system could be trained in a number of different styles, while being told which style each selection is from. It could then be asked to perform *Style Discrimination*, classifying new pieces as belonging to one of its known styles. This would probably be done by maintaining separate, well-trained context models for the various styles, and then determining the style of a new piece by selecting the model with the best prediction performance. This would extend the indirect style discrimination work discussed in the Geographical Mapping experiments.

Finally, the system can also be configured to generate music. Once trained in a certain style, Maestro could be given a starting interval and asked to predict what would come next. Its prediction for this note would then be fed back to it for use in generating the next prediction. This type of 'inspired composition' is mentioned by Conklin and Witten, and is similar to Cope's EMI work.

Computer modelling of music learning is only beginning to be explored, and prospects for the future are encouraging for both computer scientists and cognitive musicologists alike. This research has tried to explore a number of the more challenging and interesting aspects of the field.

Appendix A

Event Loop

Maestro performs eight steps when processing each musical event. Each step is presented below along with the name of the relevant Maestro functions:

1. `Read_Next_Value()` –The next input event is read in.
2. `Do_Segmentation()` – (Segmentation Stage) The segmentation modules check for a segmentation boundary at the new event and, if appropriate, suggest candidate segments for inclusion in the context model.
3. `LTM_Verify_Predictions()` and `STM_Verify_Predictions()` – (Prediction Stage) Once the new event is read, the predictions made during the previous event are compared with the actual value present in the music.
4. `Activate_Agents()` – (Modelling Stage) Segments in the context model whose first value matches the current pitch interval in the input are instantiated into listening agents.
5. `Process_Agents()` – (Prediction Stage) The currently active agents process the new event, matching the input against their internal templates and generating appropriate predictions.
6. `Parse_Agents()` – (Parsing Stage) The agents monitor and interact with any other agents with whom they are engaged in a parsing competition.
7. `Terminate_Agents()` – Agents who have mismatched the input or who have lost a parsing competition are terminated.
8. `LTM_Integrate_Predictions()` and `STM_Integrate_Predictions()` – (Prediction Stage) The predictions of the various agents are integrated to generate the probability distribution of the system as a whole.

Appendix B

Listening Agent Class

Figure B.1 shows the class declaration file for Maestro's listening agents. A listening agent has two main callable functions. In the first, `Process()`, the agent compares the input with its template and generates predictions as appropriate. This represents the agent's matching phase.

In the second function, `Parse()`, the agent monitors and interacts with the other agents involved with it in a parsing competition. This represents the agent's parsing phase. The agent keeps track of the other competitors using the two doubly-linked lists `I_Threaten` and `Threatened_By`. `Ignore_Me()` and `Persist()` are used to remove agents from these two lists respectively. If an agent loses a parsing competition, a call to `Give_Up()` is made. If it wins, the agent calls `Win()`.

```
#define La DLList <Listener *>

class Listener
{
public:
  // Variables set upon instantiation:
  int parentid;          // ID of context segment
  int id;                // ID of specific instantiation of context segment
  int sIndex;            // Starting time
  int Template[20];      // Template
  int Len;               // Length of template
  int memtype;           // Instantiated from STM or LTM context model
  int Tindex;            // Template index pointer

  // Signals indicating the state of the agent.
  int START, ON, FINISH, KILL;

  // Doubly-linked lists used for tracking other agents in Parsing Competitions
  La I_Threaten;         // List of threatened competitors
  La Threatened_By;      // List of threatening competitors

  // Callable functions
  Listener();            // Constructor
  Listener::Listener(int idnum, int st, int len, int *tlate, int mtype);
                         // Constructor with init
  void Process(void);    // Process data
  void Parse(void);      // Parse data
  void Persist(int);     // Remove a former threat from Threatened_By list
  void Ignore_Me(int);   // Remove a former competitor from I_Threaten list
  void Give_Up(void);    // Competitor wins, report termination to others.
  void Win(void);        // Claim parse
  void Clean_Up();       // Clean up tcl variables and others
};
```

Figure B.1: Maestro's listening agents are instantiations of the `Listener` class, defined in this C++ class declaration.

Appendix C

Data formats

C.1 MST

Maestro takes in data in the custom *MST* format (derived from an abbreviated form of the name Maestro), which contains information pertaining to the musical surface. Each note event is represented by a data triplet: `onset-time, pitch, duration`. This is very similar to MIDI, but no velocity information is included (see Section 3.2.1.1). Each song begins with a text string indicating the song name, followed by a series of data triplets representing the note-events of the melody. Each song is ended with the termination sequence: `-999 -999 -999`.

To prepare a corpus for experiments, multiple songs were concatenated into a single file in the desired order of presentation to Maestro. For example:

```
Song_1
<onset-time, pitch, duration>
<onset-time, pitch, duration>
...
<onset-time, pitch, duration>
-999 -999 -999
Song_2
<onset-time, pitch, duration>
.
.
.
Song_N
<onset-time, pitch, duration>
...
<onset-time, pitch, duration>
-999 -999 -999
```

```
(6
((st 12) (pitch 65) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 16) (pitch 69) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 20) (pitch 67) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 24) (pitch 69) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 28) (pitch 70) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 32) (pitch 72) (dur 8) (keysig -1) (timesig 16) (fermata 0))
((st 40) (pitch 69) (dur 4) (keysig -1) (timesig 16) (fermata 1))
((st 44) (pitch 74) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 48) (pitch 72) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 52) (pitch 70) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 56) (pitch 69) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 60) (pitch 67) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 64) (pitch 69) (dur 8) (keysig -1) (timesig 16) (fermata 1))
((st 76) (pitch 72) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 80) (pitch 74) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 84) (pitch 76) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 88) (pitch 77) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 92) (pitch 76) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 96) (pitch 74) (dur 8) (keysig -1) (timesig 16) (fermata 0))
((st 104) (pitch 72) (dur 4) (keysig -1) (timesig 16) (fermata 1))
((st 108) (pitch 69) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 112) (pitch 70) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 116) (pitch 69) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 120) (pitch 67) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 124) (pitch 67) (dur 4) (keysig -1) (timesig 16) (fermata 0))
((st 128) (pitch 65) (dur 12) (keysig -1) (timesig 16) (fermata 1)))
```

Figure C.1: Conklin and Witten's representation of a Bach Chorale used in their music prediction experiments. Chorale number 6 is shown.

As mentioned in Chapter 7, the order of songs in MST files was shuffled around using a pseudo-randomiser program called `reshuffle` written specifically for this purpose (C code). Since music from various sources was used, data had to be converted from the various original formats to the MST format, as described below.

C.2 Conklin and Witten's format

The 100 Bach Chorales used by Conklin and Witten were obtained in the form shown in Figure C.1. The shortest Chorale is shown due to space considerations. This format includes information not readily present in the

```
chorale_6
12 65 4
16 69 4
20 67 4
24 69 4
28 70 4
32 72 8
40 69 4
44 74 4
48 72 4
52 70 4
56 69 4
60 67 4
64 69 8
76 72 4
80 74 4
84 76 4
88 77 4
92 76 4
96 74 8
104 72 4
108 69 4
112 70 4
116 69 4
120 67 4
124 67 4
128 65 12
-999 -999 -999
```

Figure C.2: MST representation of Bach Chorale number 6.

musical surface: pitch signature, time signature and fermatas.

In keeping with Maestro's design specification of realistic input (Section 2.2.1), this extra information was removed from the files, yielding the corresponding MST file shown in Figure C.2.

C.3 The **kern Format

The data in the Essen Folk Song Collection was obtained in the Humdrum **kern format, which is capable of representing underlying syntactic information in the music [107]. An example is shown in Figure C.3.

The **kern representation contains phrasing information, (indicated by the '{' symbols) and bar lines ('='). Additionally, a significant amount of his-

torical and descriptive information about an individual piece is included. All these had to be removed, and the actual musical data had to be translated before processing by Maestro was possible.

The **kern data was first preprocessed with tools from the Humdrum toolkit, available with the EFSC distribution [107]. Then, a utility called `bproc` was used (written in C, roughly 200 lines of code) to convert the musical data to the MST format used by Maestro (Figure C.4).

```
!!!OTL: SO MUSS ER UNSER SCHWAGER SEIN
!!!ARE: Europa, Mitteleuropa, Deutschland
!!!SCT: E0946A
!!!YEM: Copyright 1995, estate of Helmut Schaffrath.
*kern
ICvox
Ivox
M4/4
G:
{4dd
=1
4b
4g
4a
=2
4g
4g
4g}
{4dd
=3
4dd
4dd
4dd
4dd
=4
4ee
4dd
4b}
==
!!!AGN: Tanz - Lied, Reigen
!!  Fragment
!!!ONB: ESAC (Essen Associative Code) Database:  ERK2
!!!AMT: simple quadruple
!!!AIN: vox
!!!EED: Helmut Schaffrath
!!!EEV: 1.0
-
```

Figure C.3: **kern representation of a German Folk song used in the large-scale music learning experiments. (EFSC reference number deut1527.)

```
deut1527
8 74 8
16 71 8
24 67 8
32 69 8
40 66 8
48 67 8
56 67 8
64 67 8
72 74 8
80 74 8
88 74 8
96 74 8
104 74 8
112 76 8
120 74 8
128 71 8
-999 -999 -999
```

Figure C.4: MST representation of a German Folk song used in the large-scale music learning experiments. (EFSC reference number `deut1527`.)

Appendix D

Geographical Mapping Data

Table D.1 shows the reference numbers for the songs used in the Geographical Mapping experiments reported in Chapter 10. Where many songs were available from a single style, a maximum of fifty were used. In such cases (e.g., Netherlands, Yugoslavia, etc.), instead of simply using the first fifty songs, fifty songs were selected from a distribution across the entire corpus.

Due to space constraints, the individual reference numbers for the songs from the Chinese styles are not listed. Rather, the *regular expressions* used to select the files are given instead.

COUNTRY	EFSC REFERENCE NUMBERS
Austria:	oestr010-019,030-039,050-059,070-079,090-099
Brazil:	brasil01
Canada:	canada01
China:	han0[1,2,3,4,5]?0
	natmn[0]?[1,3,5,7,9]
	shanx[1,3,5,6,7]?0
	xinhua01-10
Czech Republic:	czech01-43
Denmark:	danmark1-9
France:	france01-14,
	elsass10-19,30-39,50-79
Germany:	50 from erk1 collection
Hungary:	magyar01-45
India:	india01
Italy:	italia01-08,
	tirol01-14
Japan:	nippon01
Java:	java01
Luxemberg:	luxembrg01-08
Mexico :	mexico01-04
Netherlands:	neder010-019,030-039,050-079
Poland:	polska01-25
Romania:	romania01-28
Russia:	ussr1-36,rossiya1
Saudi Arabia:	arabic01
Sweden:	sverige01-11
Switzerland:	suisse10-19,30-39,50-59,70-89
Syria:	ashsham1
Turkey:	turkiye1
Ukraine:	ukraina01-13
United Kingdom:	england1-4
USA:	usa01-07
Yogoslavia:	jugos010-019,030-039,050-059,070-079,090-099

Table D.1: Essen Folk Song Collection reference numbers for the songs used in the Geographical Mapping experiments.

Bibliography

[1] M. Adachi and J. C. Carlsen. Measuring melodic expectancies with children. *Bulletin of the Council for Research in Music Education*, 127:1–7, 1996.

[2] E. Aldwell and C. Schachter. *Harmony and Voice Leading.* Harcourt Brace Jovanovich, 1989.

[3] J. Allen. *Natural language understanding.* Benjamin/Cummings, 1995.

[4] C. Ames. The markov process as a compositional model: A survey and tutorial. *Leonardo*, 22(2):175–187, 1989.

[5] M. Baroni. The concept of a musical grammar. *Music Analysis*, 2(2):175–208, 1983.

[6] M. Baroni, R. Dalmonte, and C. Jacoboni. Theory and analysis of European melody. In Marsden and Pople [78].

[7] T. C. Bell, J. G. Cleary, and I. H. Witten. *Text Compression.* Prentice Hall, 1990.

[8] I. Berent and A. Perfetti. An on-line method in studying music parsing. *Cognition*, 46:203–222, 1993.

[9] W. L. Berz. Working memory in music: A theoretical model. *Music Perception*, 12(3):353–364, 1995.

[10] J. J. Bharucha and P. M. Todd. Modelling the perception of tonal structure with neural nets. *Computer Music Journal*, 13(4):44–53, 1989.

[11] C. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, 1995.

[12] Brazdil, P. *et al.* Learning in distributed systems and multi-agent environments. In Y. Kodratoff, editor, *Machine Learning - European Working Session on Learning, Porto, Portugal*, pages 414–423. Springer Verlag, 1991.

[13] A. S. Bregman. *Auditory Scene Analysis: The Perceptual Organisation of Sound.* MIT Press, 1990.

[14] F. Brooks, A. Hopkins, P. Neumann, and W. Wright. An experiment in musical composition. In Schwanauer and Levitt [111].

[15] R. A. Brooks. Intelligence without reason. *Artificial Intelligence*, 47:139–160, 1991.

[16] G. J. Brown and M. Cooke. Perceptual grouping of musical sounds: A computational model. *Journal of New Music Research*, 23:107–132, 1994.

[17] J. Bryson, A. Smaill, and G. A. Wiggins. The reactive accompanist: Applying subsumption architecture to software design (RP 606). Technical report, Division of Informatics, Artificial Intelligence, University of Edinburgh, 1992.

[18] R. Byron. *Music, Culture and Experience: Selected Papers of John Blacking.* The University of Chicago Press, 1995.

[19] E. Cambouropoulos. *Towards a General Computational Theory of Musical Structure.* PhD thesis, The University of Edinburgh, 1998.

[20] E. Cambouropoulos, T. Crawford, and C. S. Iliopoulos. Pattern processing in melodic sequences: Challenges, caveats and prospects. In *Proceedings of the AISB'99 Symposium on Musical Creativity*, pages 42–47. The Society for the Study of Artificial Intelligence and Simulation of Behaviour, 1999.

[21] A. Camurri, A. Catorcini, C. Innocenti, and A. Massari. Music and multimedia knowledge representation and reasoning - the HARP system. *Computer Music Journal*, 19(2):34–58, 1995.

[22] B. Carrol-Phelan and P. J. Hampson. Multiple components of the perception of musical sequences: A cognitive neuroscience analysis and some implications for auditory imagery. *Music Perception*, 13(4):517–561, 1996.

[23] F. Chin and S. Wu. An efficient algorithm for rhythm-finding. *Computer Music Journal*, 16(2):35–44, 1992.

[24] D. D. Coffman. Measuring musical originality using information theory. *Psychology of Music*, 20:154–161, 1992.

[25] P. Collaer and A. V. Linden. *Historical Atlas of Music.* G. G. Harrap and Co. Ltd., London, 1968.

[26] D. Conklin and I. H. Witten. Prediction and entropy of music. Technical report, The University of Calgary Department of Computer Science, 1991.

[27] D. Conklin and I. H. Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24:51–73, 1995.

[28] N. Cook. *Music, Imagination and Culture.* Clarendon Press, 1990.

[29] D. Cope. An expert system for computer-assisted composition. *Computer Music Journal*, 11(4):30–46, 1987.

[30] D. Cope. Pattern matching as an engine for the computer simulation of musical style. In *Proceedings of the ICMC*, pages 288–291, 1990.

[31] D. Cope. *Computers and Musical Style.* Oxford University Press, 1991.

[32] D. Cope. Computer modelling of musical intelligence in EMI. *Computer Music Journal*, 16(2):69–83, 1992.

[33] D. Cope. A computer model of music cognition. In Schwanauer and Levitt [111].

[34] D. Cope. *Experiments in Musical Intelligence.* A-R Editions, 1996.

[35] I. Cross. Pitch schemata. In I. Deliege and J. Sloboda, editors, *Perception and Cognition of Music.* Psychology Press, 1997.

[36] J. J. Darragh and I. H. Witten. *The Reactive Keyboard.* Cambridge University Press, 1992.

[37] B. Davidson, R. P. Power, and P. T. Michie. The effects of familiarity and previous musical training on perception of an ambiguous musical figure. *Perception and Psychophysics*, 41(6):601–608, 1987.

[38] I. Deliege, M. Melen, D. Stammers, and I. Cross. Musical schemata in real-time listening to a piece of music. *Music Perception*, 14(2):117–160, 1996.

[39] I. Deliege and J. Sloboda. *Perception and Cognition of Music.* Psychology Press, 1997.

[40] Y. Demazeau. Preface. In *Proceedings Third International Conference on Multi-Agent Systems*, pages xiii–xiv, 1998.

[41] N. Dibben. The cognitive reality of hierarchical structure in tonal and atonal music. *Music Perception*, 12(1):1–25, 1994.

[42] W. J. Dowling. Context effects on melody recognition: Scale-step versus interval representations. *Music Perception*, 3(3):281–296, 1986.

[43] W. J. Dowling. Tonal strength and melody recognition after long and short delays. *Perception and Psychophysics*, 50(4):305–313, 1991.

[44] W. J. Dowling and D. L. Harwood. *Music Cognition.* Academic Press, 1986.

[45] K. Ebcioglu. An expert system for harmonising chorales in the style of J. S. Bach. In Schwanauer and Levitt [111].

[46] S. Foster, W. A. Schloss, and A. J. Rockmore. Towards and intelligent editor of digital audio: Signal processing methods. In *The Music Machine.* MIT Press, 1989.

[47] R. Frances. *The Perception of Music.* L. Erlbaum, 1988.

[48] S. Fuller. *The European Musical Heritage: 800-1750.* McGraw-Hill, 1987.

[49] D. Gang and J. Berger. Modelling the degree of realized expectation in functional tonal music: A study of perceptual and cognitive modelling using neural networks. In *Proceedings of the International Computer Music Conference*, pages 454–457, 1996.

[50] G. Gazdar and C. Mellish. *Natural language processing in Prolog: an introduction to computational linguistics.* Wokingham: Addison-Wesley, 1989.

[51] N. Griffith. Connectionist visualisation of tonal structure. *Artificial Intelligence Review*, 8(5-6):393–408, 1995.

[52] N. Griffith and P. M. Todd. *Musical Networks.* MIT Press, 1999.

[53] L. Hiller and C. Bean. Information theory analyses of four sonata expositions. *Journal of Music Theory*, 10(1):96–137, 1966.

[54] L. Hiller and R. Fuller. Structure and information in Webern's Symphonie, Op. 21. *Journal of Music Theory*, 11(1):60–115, 1967.

[55] Y. Hiraga. A computational model of music cognition based on interacting primitive agents. In *Proceedings of the International Computer Music Conference*, pages 292–295, 1993.

[56] Y. Hiraga. A cognitive model of pattern matching in music. In *Proceedings of the International Computer Music Conference*, pages 248–250, 1996.

[57] H. Honing. A microworld approach to the formalisation of musical knowledge. *Computers and the Humanities*, 27:41–47, 1993.

[58] P. Howell, I. Cross, and R. West. *Musical structure and cognition.* Academic Press, 1985.

[59] D. Huron and M. Royal. What is melodic accent? Converging evidence from musical practice. *Music Perception*, 13(4):489–516, 1996.

[60] R. Jackendoff. *Consciousness and the Computational Mind.* MIT Press, 1987.

[61] R. Jackendoff. Musical parsing and musical affect. *Music Perception*, 9(2):199–230, 1991.

[62] T. Jarvinen, P. Toivianinen, and J. Louhivuori. Classification and categorization of musical styles with statistical analysis and self-organising maps. In *Proceedings of the AISB'99 Symposium on Musical Creativity*, pages 54–57. The Society for the Study of Artificial Intelligence and Simulation of Behaviour, 1999.

[63] J. Jones, D. Scarborough, and B. Miller. GTSIM: A computer simulation of music perception. *Computers and the Humanities*, 27:19–23, 1993.

[64] M. R. Jones and M. Boltz. Dynamic attending and responses to time. *Psychological Review*, 96:459–49, 1989.

[65] M. R. Jones, L. Summerell, and E. Marshburn. Recognising melodies: A dynamic interpretation. *The quarterly journal of experimental psychology*, 39A:89–121, 1987.

[66] J. Kippen. Where does the end begin? Problems in musico-cognitive modelling. *Minds and Machines*, 2:329–344, 1992.

[67] C. Krumhansl. *Cognitive foundations of musical pitch.* Oxford University Press, 1990.

[68] C. Krumhansl. Music psychology and music theory: Problems and prospects. *Music Theory Spectrum*, 17:53–80, 1995.

[69] C. Krumhansl and P. W. Jusczyk. Infants' perception of phrase structure in music. *Psychological Science*, 1(1):70–73, 1990.

[70] S. Larson. Modelling melodic expectation: Using three musical forces to predict melodic continuations. In *Proceedings of the Annual Conference of the Cognitive Science Society*, pages 629–634, 1993.

[71] M. Leman. Artificial neural networks in music research. In Marsden and Pople [78].

[72] F. Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music.* MIT Press, 1983.

[73] C. Linster. A neural network that learns to play in different music styles. In *Proceedings of the ICMC*, pages 311–313, 1990.

[74] H. C. Longuet-Higgins and C. S. Lee. The rhythmic interpretation of monophonic music. *Music Perception*, 1(4):424–441, 1984.

[75] M. P. Lynch and R. E. Eilers. Children's perception of native and nonnative musical scales. *Music Perception*, 9(1):121–132, 1991.

[76] S. G. Makridakis, S. C. Wheelwright, and R. J. Hyndman. *Forecasting: Methods and applications.* Wiley, 1998.

[77] L. C. Manzara, I. H. Witten, and M. James. On the entropy of music: An experiment with Bach Chorale melodies. Technical report, The University of Calgary Department of Computer Science, 1991.

[78] A. Marsden and A. Pople, editors. *Computer Representations and Models in Music.* Academic Press, 1992.

[79] L. B. Meyer. Meaning and music in information theory. *Journal of Aesthetics and Art Criticism*, 15(4):412–424, 1957.

[80] L. B. Meyer. *Style and music: Theory, history and ideology.* University of Pennsylvania Press, 1989.

[81] L. B. Meyer. Commentary. *Music Perception*, 13(3):455–483, 1996.

[82] M. Minsky. Music, mind and meaning. *Computer Music Journal*, 5:28–44, 1981.

[83] M. Minsky. *The Society of Mind.* Simon and Schuster, 1987.

[84] B. A. Morrongiello. Effects of training on children's perception of music: A review. *Psychology of Music*, 20:29–41, 1992.

[85] T. Nakatani, H. G. Okuno, and T. Kawabata. Auditory stream segregation in auditory scene analysis with a multi-agent system. In *Proceedings of the conference of the American Association for Artificial Intelligence*, pages 100–107, 1994.

[86] E. Narmour. *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model.* University of Chicago Press, 1990.

[87] E. Narmour. *The Analysis and Cognition Melodic Complexity: The Implication-Realization Model.* University of Chicago Press, 1992.

[88] J. K. Ousterhout. *Tcl and the Tk Toolkit.* Addison-Wesley, 1994.

[89] R. E. Overill. On the combinatorial complexity of fuzzy pattern-matching in music analysis. *Computers and the Humanities*, 27(2):105–110, 1993.

[90] S. Phon-Amnuaisuk and G. A. Wiggins. The four part harmonisation problem: A comparison between genetic algorithms and a rule-based system. In *Proceedings of the AISB'99 Symposium on Musical Creativity*, pages 28–34. The Society for the Study of Artificial Intelligence and Simulation of Behaviour, 1999.

[91] D. Ponsford, G. Wiggins, and C. Mellish. Statistical learning of harmonic movement. *Journal of New Music Research*, 1999.

[92] B. Y. Reis. PhD Thesis Proposal and First Year Report, Cambridge University Computer Laboratory. 1997.

[93] B. Y. Reis. A multi-agent system for on-line modelling, parsing and prediction of discrete time series data. In *Intelligent Image Processing, Data Analysis and Information Retrieval*, pages 164–169. IOS Press Holland, 1999.

[94] B. Y. Reis. Simulating music learning: On-line, perceptually guided pattern induction of context models for multiple-horizon prediction of melodies. In *Proceedings of the AISB'99 Symposium on Musical Creativity*, pages 58–63. The Society for the Study of Artificial Intelligence and Simulation of Behaviour, Edinburgh, 1999.

[95] A. Rigopulos. *Growing Music from Seeds: Parametric Generation and Control of Seed-Based Music for Interactive Composition and Performance.* PhD thesis, Massachusetts Institute of Technology, September 1994.

[96] M. B. Ring. *Continual Learning in Reinforcement Environments.* PhD thesis, University of Texas, Austin, August 1994.

[97] C. Roads. *The computer music tutorial.* MIT Press, 1996.

[98] D. Rosenthal. A model of the process of listening to simple rhythms. *Music Perception*, 6(3):315–328, 1989.

[99] D. Rosenthal. Emulation of human rhythmic perception. *Computer Music Journal*, 16(1):64–76, 1992.

[100] R. Rowe. Feature classification and related response in a real-time interactive music system. In *Proceedings of the ICMC*, pages 202–204, 1990.

[101] R. Rowe. Pattern processing in music. In *Proceedings of the ICMC*, pages 60–62, 1994.

[102] R. J. Rowe. *Machine Listening and Composing: Making Sense of Music with Cooperating Real-Time Agents.* PhD thesis, Massachusetts Institute of Technology, June 1991.

[103] R. J. Rowe. Machine listening and composing with cypher. *Computer Music Journal*, 16(1):43–63, 1992.

[104] R. J. Rowe. *Interactive Music Systems: Machine Listening and Composing.* MIT Press, 1993.

[105] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 1995.

[106] H. Schaffrath. The retrieval of monophonic melodies and their variants: Concepts and strategies for computer-aided analysis. In Marsden and Pople [78].

[107] H. Schaffrath. *The Essen Folksong Collection in the Humdrum Kern Format. David Huron (ed.).* Center for Computer Assisted Research in the Humanities, Menlo Park, CA, 1995.

[108] R. J. Schalkoff. *Pattern recognition: statistical, structural and neural approaches.* Wiley, 1992.

[109] E. G. Schellenberg. Simplifying the implication-realization model of melodic expectancy. *Music Perception*, 14(3):295–318, 1997.

[110] M. A. Schmuckler. Expectation in music: Investigation of melodic and harmonic processes. *Music Perception*, 7(2):109–150, 1989.

[111] S. M. Schwanauer and D. A. Levitt, editors. *Machine Models of Music.* MIT Press, 1993.

[112] M. L. Serafine. *Music as Cognition: The Development of Thought in Sound.* Columbia University Press, 1988.

[113] M. L. Serafine, N. Glassman, and C. Overbeeke. The cognitive reality of hierarchic structure in music. *Music Perception*, 6(4):397–430, 1989.

[114] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423 and 623–656, 1948.

[115] J. A. Sloboda. *The Musical Mind: The Cognitive Psychology of Music.* Clarendon Press, 1985.

[116] J. D. Smith. The place of musical novices in music science. *Music Perception*, 14(3):227–262, 1997.

[117] J. P. Swain. The need for limits in hierarchical theories of music. *Music Perception*, 4:121–148, 1986.

[118] A. S. Tanguiane. *Artificial Perception and Music Recognition.* Springer-Verlag, 1993.

[119] J. M. Thomassen. Melodic accent: Experiments and a tentative model. *Journal of the Acoustical Society of America*, 11(6):1596–1605, 1982.

[120] W. Thomson. Functional ambiguity in musical structures. *Music Perception*, 1:3–27, 1983.

[121] J. T. Titon. *Worlds of Music: An Introduction to the Music of the World's Peoples.* Schirmer Books, 1992.

[122] N. Todd. The auditory 'primal sketch': A multiscale model of rhythmic grouping. *Journal of New Music Research*, 23:25–70, 1994.

[123] N. Todd. Toward a theory of the central auditory system i, ii, iii, iv. In *Proceedings of the ICMPC*, pages 173–196, 1996.

[124] G. Weiss. Adaptation and learning in multi-agent systems: Some remarks and a bibliography. In G. Weiss and S. Sen, editors, *Adaptation and Learning in Multi-Agent Systems: IJCAI'95 Workshop, Montreal, Canada*, pages 1–21. Springer, 1996.

[125] M. D. Westhead and A. Smaill. Automatic characterization of musical style. In M. Smith, A. Smaill, and G. Wiggins, editors, *Music Education: An Artificial Intelligence Perspective, Edinburgh 1993*, pages 157–170. Springer Verlag, 1994.

[126] G. Widmer. The importance of basic musical knowledge for effective learning. In M. Balaban, K. Ebcioglu, and O. Laske, editors, *Understanding Music with AI: Perspectives on Music Cognition.* AAAI Press, 1992.

[127] G. Widmer. Qualitative perception modelling and intelligent musical learning. *Computer Music Journal*, 16(2):51–68, 1992.

[128] G. Widmer. Modelling the rational basis of musical expression. *Computer Music Journal*, 19(2):76–96, 1995.

[129] G. Widmer. What is it that makes it a Horowitz? Empirical musicology via machine learning. In *ECAI '96; 12th European Conference on Artificial Intelligence*, pages 458–462, Chichester - New York - Brisbane, August 1996. Wiley.

[130] G. Wiggins, E. Miranda, A. Smaill, and M. Harris. A framework for the evaluation of music representation systems. *Computer Music Journal*, 17(3):31–42, 1993.

[131] I. H. Witten and D. Conklin. Modelling music: Systems, structure and prediction. *Interface*, 19:53–66, 1990.

[132] I. H. Witten, L. C. Manzara, and D. Conklin. Comparing human and computational models of music prediction. *Computer Music Journal*, 18(1):70–80, 1994.

[133] H. Zielinska and K. Miklaszewski. Memorising two melodies of different style. *Psychology of Music*, 20:95–11, 1992.