Machine Learning and Sound Design

Eduardo Reck Miranda
SONY CSL - Paris
6 rue Amyot
75005 Paris - France

Abstract

In this paper I discuss the role of Machine Learning (ML) in sound design. I focus on the modelling of a particular aspect of human intelligence which is believed to play an important role in musical creativity: the Generalisation of Perceptual Attributes (GPA). By GPA I mean the process by which a listener tries to find common sound attributes when confronted with a series of sounds. The paper introduces the basics of GPA and ML in the context of ARTIST, a prototype case study system. ARTIST (Artificial Intelligence Sound Tools) is a sound design system that works in co-operation with the user, providing useful levels of automated reasoning to render the synthesis tasks less laborious (tasks such as calculating an appropriate stream of synthesis parameters for each single sound) and to enable the user to explore alternatives when designing a certain sound. The system synthesises sounds from input requests in a relatively high-level language; for instance, using attribute-value expressions such as "normal vibrato", "high openness" and "sharp attack". ARTIST stores information about sounds as clusters of attribute-value expressions and has the ability to interpret these expressions in the lower-level terms of sound synthesis algorithms. The user may, however, be interested in producing a sound which is "unknown" to the system. In this case, the system will attempt to compute the attribute values for this yet unknown sound by making analogies with other known sounds which have similar constituents. ARTIST uses ML to infer which sound attributes should be considered to make the analogies.

1 Introduction

Recent studies in acoustics, psychoacoustics, psychology and cognitive sciences have vastly expanded our knowledge of the nature and perception of sounds and music. The sound domain of Western music is no longer demarcated by the boundaries of traditional acoustic instruments. Nowadays, composers have the opportunity to create music with an infinite variety of sounds, ranging from "natural sounds" (those produced by acoustic devices and different sorts of mechanical excitation; such as the sounds produced by blowing a pipe (Rossing, 1990)) to synthesised, "artificial sounds" (those sounds that cannot be produced by acoustic devices; such as the sounds produced by a cellular automata synthesiser (Miranda et al., 1992; Miranda, 1995b)).

Computer technology offers the most detailed control of the internal parameters of synthesised sounds, which enables composers to become more ambitious in their quest for a more effective use of sound synthesis technology. In this case however, the task of sound composition becomes more complex. A composer can set the parameters for the production of an immeasurable variety of sounds, but this task is still accomplished unnaturally by inputting streams of numerical data specified manually (as in the case of the Csound score files, for example (Vercoe, 1991)). Even if composers know the role played by each single parameter for synthesising a sound, it is both very difficult and tedious to ascertain which values will synthesise the sound they want to produce. Moreover, composers often need to master a sound synthesis programming language in order to communicate with the computer (as in the case of the programming language CLM (Common Lisp Music), for example (Schottstaedt, 1992; 1994)). Even if they master this language, the design of an instrument is not a straightforward task. In such a situation, higher processes of inventive creativity and abstraction become subsidiary to time consuming, non-musical tasks. Composers need a better working environment.

It seems that the interdisciplinary knowledge we have about the nature and perception of sounds (that is, acoustics, psychoacoustics, psychology, etc.) has not been taken into account by sound synthesis software engineers. Better sound design systems can be provided if we devise ways for including this knowledge in a sound design software. I believe that this situation can be improved by combining computer sound synthesis technology with Artificial Intelligence (AI) techniques. AI techniques are aimed here to help us to devise sound synthesis systems that take into account the interdisciplinary knowledge mentioned above.

I have been working on an Artificial Intelligence-based approach for improving sound design systems. In order to test this approach, I have designed ARTIST (Artificial Intelligence Sound Tools): a prototype case study system that allows the design of sounds by thinking in terms of user-customised qualitative descriptions rather than in terms of numerical streams. Moreover, this system also works in co-operation with the user by providing support for the exploration of ideas.

In order to design sounds, composers often explore a variety of possible solutions by trying out possibilities within a certain personal style or idiom. ARTIST maintains user-customised descriptive information about known sounds on a knowledge base. The user can request sounds either by using its name (if the sound is known) or by using a number of sound descriptors (if the sound is unknown). In the latter case, the system attempts to create the sound by making analogies with known sounds. To accomplish this, ARTIST uses machine learning techniques (ML) to infer which sound attributes should be considered to make the analogies. ML is the subfield of Artificial Intelligence that studies the phenomenon of learning by developing computer programs that can learn.

In this paper I will focus on the ML engine of ARTIST. I begin with some background concepts, including a brief presentation of the architecture of the system and a discussion about its knowledge representation technique. Next, I introduce the basics of ML and discuss the ML engine used in the system. Then, I present an example operation and end with some final remarks and envisage further work. More details of ARTIST's underlying philosophy, architecture and functioning can found in (Miranda 1994a; 1994b; 1995a; 1995c; Miranda et al., 1993).

2 Background Concepts

2.1 Sound Synthesis as Knowledge-Based Design Problem

Design is a complex kind of intelligent behaviour. It is concerned with engaging in cognitive and physical acts in order to establish the suitability and effectiveness of our creations prior to actually constructing them. In attempting to solve design problems, designers explore the space of possible solutions by trying out possibilities and investigating their consequences.

One cannot hope to fully understand design by adopting a single perspective on its study, but we must combine the perspectives of many different disciplines. Nevertheless, I am interested for present purposes, in a limited aspect of design: design as an explicitly knowledge-based kind of intelligent behaviour. It is therefore assumed that it involves the explicit organisation, application and generation of knowledge.

Artificial Intelligence is a science which aims at understanding intelligent behaviour and how it might be artificially created to serve specific goals (Luger and Stubblefield, 1989). In this context, in order to understand design as a kind of intelligent behaviour one needs ways to describe and express aspects of the behaviour being investigated: how one thinks this behaviour can be modelled and how one thinks it can possibly be aided, or even simulated by a computer.

As I am primarily interested in electroacoustic music (Emmerson, 1994), this work is based upon the assumption that composition is not only considered to be the combination of pre-existing sounds; it also involves an effort to elaborate the sound material (that is, creating the sounds themselves, rather than merely composing with existing sounds).

It is generally agreed nowadays that there are no fixed boundaries between the design of sound and the composition of music. A composer might either think of an evolving single sound that is in itself a piece of music, or of a combination of several discrete sound events. When synthesising sounds to be used in a piece of music, musicians have an intuition about the possibilities of the organisation of these sounds into a musical structure. In attempting to obtain the desired sound, composers explore a variety of possible solutions, trying out those possibilities within their personal aesthetic (Roozendal, 1993). This process of exploration frequently results in inconsistencies between the composer's best guess at a solution and the formulation of the requirement. If no solution meets the requirement, then this requirement either has no solution at all, or it must be redefined. Sound design is seen in this context as a problem which demands, on the one hand, clarification of the requirement and, on the other hand, provision of alternative solutions. For example, suppose that a composer wants to synthesise a "high-pitch sound". In order to produce this sound, the system might need the expression "high-pitch sound" to be clarified by enunciating that "high-pitch" actually means a fundamental frequency above a certain threshold. If the system still does not understand the clarification, then some sound at least should be produced, which would give some chance that the sound produced may satisfy the user's requirement.

2.2 Representing Knowledge of Sounds

Knowledge representation is a fundamental aspect of ARTIST and knowledge-based systems in general. The representation technique defines the nature of the information and the mechanism for processing it.

It is generally assumed that intelligent activity is mediated by internal representations. There is no consensus, however, on what these representations are; some regard them as neurophysiological states, whilst others may define them as symbols or even images. A system such as ARTIST needs a technique that supports the representation of both sound description and synthesis. For the present purposes, I speculate that descriptions of sounds are based upon a kind of sonic image of sounds' contours in a phenomenal field, which helps our mind to identify their attributes. On a computer, this phenomenon can roughly be simulated using *schemas* (Miranda, 1994a; 1994b; 1995a; 1995c).

Figure 1: *ASS is a multi-levelled structure which mediates sound descriptions and their corresponding synthesis parameters.*

[FIGURE 1 TO BE PLACED HERE]

The schema devised for ARTIST is called Abstract Sound Schema (ASS). ASS is a multi-levelled structure aimed at mediating sound descriptions - that is sound attributes, created from a user-defined vocabulary of descriptors - and their respective synthesis parameters (att=attack, amp=amplitude, sust=sustain, rel=release, f0=frequency). The role of the ASS is twofold: it embodies a multi-levelled representation of the signal processing of an instrument and also provides an abstraction to represent sounds (Figure 1).

The ASS consists of: nodes (black squares), slots (black circles) and links. Nodes and slots are components and links correspond to the relations between them. Both nodes and slots on the ASS are labelled. Slots are grouped "bottom up" into higher level nodes, which in turn are grouped into higher level nodes and so on until the top node. Each slot has a label and accommodates either a sound synthesis datum or a pointer towards a procedure to calculate a sound synthesis datum; this information is stored in a knowledge base. Higher level nodes correspond to the modules and sub-modules of the signal processing architecture; they also have labels (Figure 2).

Figure 2: *The ASS representation of the instrument shown in Figure 2.*

[FIGURE 2 TO BE PLACED HERE]

ASS enables the organisation of knowledge of sounds based upon the signal processing model that produces them, (e.g., a Physical Model algorithm (Borin et al., 1992; Smith III, 1996)). A sound event is represented here in terms of the various perceptual features which contributes to its identity. These features must however be tied to the signal processing model in some way. It is assumed that each descriptive attribute (identified in a sound event produced by an instrument) is caused by a certain component, or a group of components, of this instrument. Let us take as an example the instrument shown in Figure 1. If we tie an attribute called "pitch" to the oscillator component, the attribute value "high-pitch" could be made to correspond to a high frequency value; for example, f0=1760 Hz.

2.3 The Architecture of the System and Basic Functioning

Figure 3 illustrates the main modules of the system's architecture and their connection. ARTIST synthesises sounds from requests in a relatively high-level language, via the *User Interface*.

Information about sounds and attributes are stored in the *Knowledge Base* as clusters of expressions (Figure 4). The *Assemblage Engine* consults the *Knowledge Base* in order to compute the slot values of either the whole ASS structure, or single nodes (i.e., the *Assemblage Engine* "assembles" the sound). An assemblage of the whole schema corresponds to a sound, whereas assemblages of single nodes correspond to sound attributes.

Figure 3: *The main modules of ARTIST.*

[FIGURE 3 TO BE PLACED HERE]

In order to synthesise a sound, the user may enter either the name of the sound, or a list of attribute values. If the list is incomplete (i.e., there are not enough attributes to assemble the sound) or obscure (i.e., the system does not recognize some of the attribute-values), ARTIST will attempt to guess missing or inconsistent information using the rules produced by the *Machine Learning Engine*, stored in *Induced Rules.*

Figure 4: *The Assemblage Engine firstly collects the appropriate slot values in order to assemble the desired sound and then activates the synthesis algorithm.*

[FIGURE 4 TO BE PLACED HERE]

3 Machine Learning

Machine learning (ML) is a major sub-field of AI, with its own various branches. Perhaps the most popular current debate in ML, and in AI in general, is between the sub-symbolic and the symbolic approaches. The former, also known as connectionism or neural networks, is inspired by neurophysiology; it intends to provide mechanisms so that the desired computation may be achieved simply by repeatedly exposing the system to examples of the desired behaviour. As the result of learning, the system records the "behaviour" in a network of single processors (metaphorically called "neurones").

ARTIST currently uses the more traditional ML symbolic approach (Winston, 1984). Several algorithms for symbolic learning have been employed in AI systems. These range from learning by being told to learning by discovery (Bratko, 1990; Carbonell, 1990). In the former case, the learner is told explicitly what is to be learned by a "teacher". In learning by discovery, the system automatically discovers new concepts, merely from observations, or by planning and performing experiments in the domain. Many other techniques lie between these two extremes. The criteria for selecting a machine learning technique depends upon many factors, including its purpose and the representation scheme at hand. In this work, the selection was inspired by a psychoacoustic speculation.

3.1 A Psychoacoustic Speculation: The Generalisation of Perceptual Attributes

It is believed that, when people listen to several distinct sound events, they tend to characterise them by selecting certain sound attributes which they think are important. When listening to several distinct sound events, it seems that the human mind prioritises the selection of certain attributes which are more important in order to make distinctions among them (Miranda, 1994b).

If one carefully listens to a series of sound events, there will probably be a large number of possible intuitive generalizations. It is therefore essential to select those generalizations we believe to be appropriate. These depend upon several factors such as context, sound complexity, duration of events, sequence of exposure and repetition, which make a great variety of combinations possible. Humans, however, are able to make generalizations very quickly; perhaps because we never evaluate all the possibilities. We tend to limit our field of exploration and resort to some heuristic. I believe that this plays an important role in imagination and memory when creating sounds and composing with them.

The purpose of ML in ARTIST is to induce general concept descriptions of sounds, from a set of examples. The ML technique selected for our investigation is therefore the *inductive learning technique* (IML). The benefit of being able to induce general concept descriptions of sounds is that the machine can automatically use induced concept descriptions to identify unknown sounds and to suggest missing attributes of an incomplete sound description.

IML provides ARTIST with the ability to make generalizations in order to infer which attributes are "more distinctive" in a sound. The term "more distinctive" in this case does not necessarily refer to what humans would perceive to be the most distinctive attribute of a sound. Current ML techniques are not yet able to mimic all the types of heuristics used by humans. Nevertheless, I propose that one kind of heuristic might use information theory to make generalizations. The algorithms used in ARTIST thus use information theory to pursue this task. Once the generalizations have been learned, the user may use the descriptive rules to specify new sounds, different from those that were originally picked out as typical of the sounds that the system already "knows".

The aim of inducing rules about sounds is to allow the user to explore further alternatives when designing particular sounds. The user could ask the system, for example, to "play something that sounds similar to a vowel" or even "play a kind of dull, low pitched sound". In these cases the system would consult induced rules to infer which attributes would be necessary to synthesise a vowel-like sound, or a sound with dull colour attribute.

3.2 The Inductive Machine Learning Engine

The target of IML in ARTIST is to induce concepts about sounds represented in a training set. The training set can be either new data input by the user, or automatically inferred by the system, from its own knowledge base.

Inductive learning can be either incremental, modifying its concepts in response to each training example, or single trial, forming concepts once in response to all data. A classic example of incremental inductive learning is a program called ARCHES (Winston, 1985). ARCHES is able to learn the structural description of an arch from examples and counter-examples supplied by a "teacher". The examples are processed sequentially and ARCHES gradually updates its current definition of the concept being learned by enhancing either the generality or the specifity of the description of an arch. It enhances the generality of the description in order to make the description match a given positive example, or the specifity in order to prevent the description from matching a counter-example.

The Iterative Dichotomizer 3 (ID3) algorithm is a classic example of single trial inductive learning (Quinlan, 1986). ID3 induces a decision tree from a set of examples of objects of a domain; the tree classifies these objects according to their attributes. Each example of the training set is described by a number of attributes. The ID3 algorithm builds a decision tree by measuring all the attributes in terms of their effectiveness in partitioning the set of target classes; the best attribute (from a statistical standpoint) is then elected as the root of the tree and each branch corresponds to a partition of the classifications (i.e., values of this attribute). The algorithm recurs on each branch in order to process the remaining attributes, until all branches lead to single classification leaves.

In principle, any technique that produces classificatory rules based upon attributional descriptions could be useful. Ideally the system should use various IML algorithms in order to provide more than one classificatory possibility. The ability to have more than one classificatory possibility is useful in a situation where, for example, the user inputs a request to produce a sound and the system must check whether it knows a sound that matches this request. Therefore by having more than one classificatory rule, the system has a greater chance of finding a matching sound and indeed of finding more than one sound which satisfies the requirement. To this end, I arbitrarily implemented two single trial IML algorithms: the Induction of the Shortest Concept Description (ISCD) and the Induction of Decision Trees (IDT) (Dietterich and Michalski, 1981; Bratko, 1990).

The ISCD algorithm aims to induce the shortest description(s), that is, the smallest set(s) of attribute values of a sound (or class of sounds) which can differentiate it from the others in the training set. The IDT algorithm also induce classificatory rules, but not necessarily the most succinct ones. In this paper I focus only on the latter algorithm; the former has been discussed in (Miranda, 1994b; 1996).

ARTIST's IDT algorithm is an adapted version of a Quinlan's-like algorithm described in (Bratko, 1990). The result of the learning is represented in the form of a Decision Tree (DT), where internal nodes are labelled with attributes and branches are labelled with attribute values (note, however, that the DT is not the same as the ASS representation discussed earlier). The leaves of the tree are labelled with sound classes. To classify a sound event, a path in the tree is traversed, starting at the root node and ending at a leaf. The IDT algorithm (see appendix) proceeds by searching, at each non-terminal node, for the attribute whose values provide the best discrimination among the other attributes, that is, the Most Informative Attribute (MIA); the formula for the selection of the MIA has been explained in (Miranda, 1994b).

Figure 5: *An example DT induced from an hypothetical training set.*

[FIGURE 5 TO BE PLACED HERE]

Figure 5 shows an example DT induced from an hypothetical training set such as follows:

Sound Name: *dull*
Sound Attributes:
openness = *wide*
pitch = *high*
vibrato rate = *default*

Sound Name: *wobbly*
Sound Attributes:
openness = *wide*
pitch = *low*
vibrato rate = *fast*

etc ...

Once the decision tree is induced, to identify a sound a path is traversed in the tree, starting at the root (top sound attribute) and ending at a leaf. One follows the branch labelled by the attribute value at each internal node. For example, a sound described by "wide openness, low pitch and fast vibrato rate" is classified, according to this tree, as *wobbly*.

4 An Example Operation

Suppose that ARTIST holds knowledge of a synthesiser that has three components (*Vibrato Source*, *Pulse Generator* and *1st Formant*) and five parameters (rate, width, f0, f1 and bw1); its ASS representation is shown in Figure 6 and the *Knowledge Base* is partially illustrated in Figure 7. Note that the *Knowledge Base* also contains a 'dictionary' of nodes, in addition to clusters of slots and attributes. This dictionary relates user-specified labels, or attributes, with the components of the instrument; for example, the attribute "openness" has been attached to the *1st Formant* component and values for this attribute (e.g., "wide") will depend upon the values of *f1* and *bw1*. Also, assume that the *Machine Learning Engine* automatically made a training set out of the *Knowledge Base* _and induced the DT illustrated in Figure 5.

Figure 6: *The ASS representation of the example operation instrument.*

[FIGURE 6 TO BE PLACED HERE]

If the user requests a sound called "wobbly", ARTIST immediately retrieves the slots from the *Knowledge Base*, assemble the ASS and synthesises the sound; for example, ARTIST knows that the "wobbly" sound has "wide openness", that openness corresponds to the "1st Formant" component, that the parameters for this component should value f1=650 Hz and bw1=65 Hz, and so forth. In this case, there was no need to consult its induced rules.

Conversely, if the user requests a sound by using attributional descriptions, such as "wide openness and medium pitch", ARTIST will realise that this description is incomplete (i.e., it does not mention the vibrato rate) and will consult its induced rules in order to deduce suitable values for the missing information. The DT (Figure 5) suggests that the "female vowel" sound is a good candidate to fulfill the requirement, regardless of its vibrato. ARTIST would then synthesise the "female vowel" sound. In this case, the user might or might not be satisfied. If not, the system may either suggest other sounds (if there are any other possibilities) or ask for additional information to enhance the request.

Figure 7: *An example of a Knowledge Base.*

[FIGURE 7 TO BE PLACED HERE]

5 Final Remarks and Further Work

At the beginning of this century, Stravinsky envisaged the type of working environment where he could give sound descriptions to an engineer (such as, "something electronic, kind of middle range, bassoon-trombone like") in order to manufacture electronic sounds. A few years later, Boulez made the avant-garde composer's dream possible: he created IRCAM, a research centre full of engineers, based in Paris. IRCAM's engineers designed a large computer music system, called 4X, and provided support to Boulez (and to composers who could afford to visit Paris) to "manufacture" the sounds for his compositions. Fortunately, modern computer technology now enables the simulation of 4X-like systems on smaller personal computers. Most composers can take advantage of this technology, but they still need, however, better ways to operate such machines. New systems, such as ARTIST, are therefore a natural progression.

In this paper I have discussed the role of Inductive Machine Learning (IML) in ARTIST. I have focused on the modelling of a particular aspect of human intelligence which is believed to play an important role in musical creativity: the Generalisation of Perceptual Attributes (GPA).

At the moment, the attribute-value pairs for sound description are specified manually. I plan to automatize this task by adding the support of a sub-symbolic level to the symbolic IML level of ARTIST. Neural networks technology is suitable for this task (Forrest et al., 1987). I propose that a neural network based upon auditory modelling techniques has great potential for raising new paradigms for sound representation. In addition, this would enable the creation of a more perceptually-oriented tool for sound analysis and therefore facilitate the definition of sound descriptors for a sound. The sub-symbolic level would then be aimed at the identification of prominent classificatory features in input samples of sounds and provide ways of referring them using symbols to be processed at the symbolic IML level.

References

[**Emmerson, 1994**] Emmerson, S. (1994), (Ed.), Contemporary Music Review, issue on "Timbre Composition in Electroacoustic Music", 10:2, Harwood Academic Publishers (ISBN 3-7186-5572-1).

[**Bratko, 1990**] Bratko, I. (1990), Prolog Programming for Artificial Intelligence, Addison Wesley (ISBN 0-201-41606-9).

[**Borin et al., 1992**], Borin, G., De Poli, G. and Sarti, A. (1992), "Algorithms and Structures for Synthesis Using Physical Models", Computer Music Journal, 16:4, pp. 30-42, MIT Press (ISSN 0148-9267).

[**Carbonel, 1990**] Carbonell, J.G. (1990), "Introduction: Paradigms for Machine Learning", Machine Learning: Paradigms and Methods, Carbonell, J. G. (Ed.), MIT Press/ Elsevier.

[**Dietterich and Michalski, _1981**] Dietterich, T. and Michalski, R. (1981), "Inductive Learning of Structural Descriptions", Artificial Intelligence, Vol. 16, Elsevier.

[**Forrest et al., 1987**] Forrest, B.M., Roweth, D., Stroud, N., Wallace, D.J. and Wilson, G.V. (1987), Neural Networks Models, Physics Department pre-print 87/419 (ECSP-TR-11), University of Edinburgh.

[**Luger and Stubblefield, 1989**] Luger, G. F. and Stubblefield, W. A. (1989), Artificial Intelligence and the Design of Expert Systems, Benjamin/Cummings Publishing (ISBN 0-80530139-9).

[**Miranda, 1994a**] Miranda, E.R. (1994a), "From Symbols to Sound: AI-based Investigation of Sound Synthesis.", Contemporary Music Review, 10:2, pp. 211-232, Harwood Academic Publishers (ISBN 3-7186-5572-1).

[**Miranda, 1994b**] Miranda, E.R. (1994b), Sound Design: An Artificial Intelligence Approach, PhD thesis, University of Edinburgh.

[**Miranda, 1995a**] Miranda, E.R. (1995a), "An Artificial Intelligence Approach to Sound Design", Computer Music Journal, 19:2, pp. 59-75, MIT Press (ISSN 0148-9267).

[**Miranda, 1995b**], Miranda, E.R. (1995b), "Granular Synthesis of Sounds by Means of a Cellular Automaton", Leonardo, 28:4, pp. 297-300 (ISSN 0024-094X).

[**Miranda, 1995c**] Miranda, E.R. (1995c), "ARTIST: An Artificial Intelligence-based System for Sound Design", Proceedings of Deuxiemes Journees d'Informatique Musicale JIM'95, Paris (France), pp. 109-122, Institut Blaise Pascal.

[**Miranda, 1996**] Miranda, E.R., (1996), "Machine Learning of Sound Attributes: Computer-Assistance in Concept Formation and Musical Invention", Proceedings of the 1996 International Computer Music Conference, Hong Kong, pp. 479-482, ICMA (ISBN 962-85092-1-7).

[**Miranda et al., 1992**] Miranda, E.R., Nelson, P. and Smaill, A. (1992), "ChaOs: a Model for Granular Synthesis by Means of Cellular Automata", Annual Report and Project Directory 91-92, pp. 153-156, Edinburgh Parallel Computing Centre (EPCC).

[**Miranda et al., 1993**] Miranda, E.R., Smaill, A. and Nelson, P. (1993), "A Symbolic Approach for the Design of Intelligent Musical Synthesizers", Proceedings of the X Reunion Nacional de Inteligencia Artificial, Mexico City (Mexico), Grupo Noriega Editores (ISBN 968-18-4730-X).

[**Quinlan, 1986**] Quinlan, J. R. (1986), "Induction of decision trees", Machine Learning, 1:1.

[**Roozendal, 1993**] Roozendal, R. (1993), "Psychological analysis of musical composition: composition as design", Contemporary Music Review, 9:1&2, pp. 311-324, Harwood Academic Publishers (ISBN 3-7186-5572-1).

[**Rossing, 1990**] Rossing, T.D. (_1990), The Science of Sound (2nd edition), Addison-Wesley (ISBN 0-201-15727-6).

[**Schottstaedt, 1992**] Schottstaedt, W. (1992), Common Lisp Music Documentation, available via Internet ftp from the clm directory on the host machine ccrma-ftp.Stanford.edu.

[**Schottstaedt, 1994**] Schottstaedt, W. (1994), "Machine Tongues XVII: CLM: Music V Meets Common Lisp", Computer Music Journal,18: 2, pp. 30-37, MIT Press (ISSN 0148-9267).

[**Smaill et al., 1994**] Smaill, A., Wiggins, G. and Miranda, E.R. (1994), "Music Representation -

between the Musician and the Computer", Music Education: An Artificial Intelligence Approach, Smith et al. (Eds), Springer-Verlag (ISBN 3-540-19873-3, ISBN 0-387-19873-3).
[**Smith III, 1996**] Smith III, J.O. (1996), "Physical Modelling Synthesis Update", Computer Music Journal, 20:2, pp. 44-56, MIT Press (ISSN 0148-9267).
[**Vercoe, 1991**] Vercoe, B. (1991), Csound Manual, available via Internet ftp from the music directory on the host machine media-lab.mit.edu.
[**Winston, 1984**] Winston, P. (1984), Artificial Intelligence (2nd edition), Addison-Wesley.
[**Winston, 1985**] Winston, P. (1985), "Learning Structural Descriptions from Examples", Readings in Knowledge Representation, Brachman, R. and Levesque, H. (Eds.), Morgan Kaufmann Publishers (ISBN 0-934613-01-X).

Appendix: The IDT Algorithm

To construct a decision tree DT from a training set TSet do:

1. If **TSet** is empty then **DT** is a single-node tree labelled **nul**l
2. Else
   2.1. If all the examples in **TSet** belong to the same sound class **SOUND_CLASS**
   2.2. Then **DT** is a single-node tree labelled **SOUND CLASS**
   2.3. Else select the most informative attribute **MIA**
      2.3.1. If there is no **MIA** to choose
      2.3.2. Then **DT** is a single-node tree with the list of the conflicting examples
      2.3.3. Else
         2.3.3.1. From the **MIA** obtain its attribute values **atv(1), atv(2), ..., atv(n)**;
         2.3.3.2. Partition **TSet** into **TSet(1), TSet(2), ..., TSet(n)**, according to the attribute values atv of **MIA**;
         2.3.3.3. Construct recursively sub-decision-trees **ST(1), ST(2), ..., ST(n)** for **TSet(1), TSet(2), ..., TSet(n)**;
         2.3.3.4. The result is the tree **DT** whose root **MIA** and whose sub-decision-trees are **ST(1), ST(2), ..., atv(2), ..., atv(n)**.

Figure 8: Each time a new MIA is selected the algorithm constructs recursively sub-decision-trees ST for each attribute of the MIA.

[FIGURE 8 TO BE PLACED HERE]

Each time a new MIA is selected, only those attributes which have not yet been selected in previous recursion (that is, used in the upper parts of the tree) are considered (Figure 8).

When the available attributes are insufficient to distinguish between classes of sound examples (that is, sound examples that belong to different classes may have exactly the same attributes) then we say that these are conflicting examples. If the algorithm cannot find a new MIA, then it records a list of conflicting examples together with the number of occurrences in TSet of each element of the conflicting list. This information is used as a weight if a selection among them is eventually required.