CHAPTER 1
INTRODUCTION AND BACKGROUND

Inductive learning can be used to perform an expert skill using nearest neighbor (NN) pattern recognition. This is demonstrated through a sound equalization expert system which learns to proficiently adjust the timbres (sound qualities) of brightness, darkness, and smoothness in a context-dependent fashion, creating an intelligent computer interface. This is innovative in that it applies the established nearest-neighbor technique to the new application area of performing a skillful perceptual task. This combination has been made possible through advances in computer memory and processor technology, making previously intractable problems now feasible. This work also demonstrates a computer-human interaction (CHI) paradigm where the computer is used as a tool to sense, process, and act in helping the user perform a perceptual task.

The expert system developed here for doing sound equalization is an example of capturing the valuable commodity of human expertise using a computer. Computer learning is needed to help overcome the knowledge acquisition bottleneck for these systems. Human expertise can be separated into expert knowledge and expert skill. Expert knowledge consists of that which you know how to do, such as knowing when medical symptoms indicate a heart attack or who composed a particular piece of music. Expert skill consists of that which you are able to do, such as being able to perform heart bypass surgery or to play a piece of music. Skills as such do not constitute what we think, but rather what we are. An aging professional athlete may still know what to do,
but his or her body may no longer be able to execute the action. In our expert system the skill consists of changing qualities (timbres) of sounds through equalization.

Learning an expert skill involves improving performance through repetition of some perceptual task. Since the interpretation of sensory input for each person may be subjectively defined and context dependent, in many domains it is computationally impossible to develop a universal set of expert rules for a given perceptual task. Instead we have chosen to use the accumulation of experience by way of inductive inference using nearest neighbor pattern matching. Reasons for choosing nearest neighbor over other inductive methods are given later in section 3.1.

1.1 A Perceptual Task

The work presented here is unique in the combination of technologies used in learning a perceptual task. For the sake of the discussion here we define machine learning as “getting machines to improve their performance with experience.” We define a perceptual task as a task where sensory input is processed to appropriately perform some action, e.g. riding a bike or vocal harmonization. We differentiate between sensing and perceiving in that perceiving takes the additional step of incorporating the sensory input into some sort of usable representation. Perceiving is not just observing, but additionally apprehending. We also differentiate between a perceptual task and an expert, or skillful, perceptual task. Many people can drive a car, but few have the skill to drive in a race. Many people can tell which of two equalizations for a piece of music they prefer, but few have the skill to isolate which frequency bands cause the differences. Webster’s ninth College Dictionary defines skill as “the ability to use one's knowledge effectively and readily in execution or performance; dexterity or coordination especially in the execution of learned physical tasks; a learned power of doing something competently: a developed aptitude or ability (e.g. language skills).”
Early A.I. researchers were optimistic about implementing computer systems for simple sensory mundane tasks such as vision, locomotion or understanding speech, thinking that these simple tasks would be solved more easily than the “expert” tasks such as medical diagnosis, engineering design, or even a formal task such as playing chess. The reasoning behind this was that since sensory mundane tasks were abilities that were acquired first in human development (Rich and Knight 1991), they would be simpler and easier to computerize. The opposite has proven to be true. Sensory-rich mundane tasks have proven to be extremely difficult, while there have been notable successes among specific “expert” tasks (microworlds), where a system contains enough knowledge in a limited domain to perform expertly.

In this research the computer is used as a tool to aid the user in both perceiving and performing a task, where the system learns an expert skill. Different combinations of a human and a computer perceiving and performing a task are shown in Figure 1. Note that there is a significant distinction between the computer doing the perceiving to perform a task and the human doing the perceiving and then telling the computer what to do.

<table>
<thead>
<tr>
<th>Human perceives</th>
<th>Both perceive</th>
<th>Computer perceives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vocal harmonizing</td>
<td>Eq (This research)</td>
<td>Speech Recognition</td>
</tr>
<tr>
<td>Digital Sound Effects</td>
<td></td>
<td>Autonomous Vehicle Navigation</td>
</tr>
</tbody>
</table>

Figure 1: Perceiving and Executing Task: Human or Computer
1.2 Sound Equalization

The perceptual task of sound equalization is a particularly rich domain in which to develop an expert system. It is used in public address systems, recording studios, movie theatres and stereo systems. At a very basic level it is encountered on home stereo systems as the treble and bass tone controls. These act as amplifiers and filters changing the amount of energy in different frequency bands. The greater the number of discrete bands, the finer the control that can be applied to a sound. Representing sounds as energy per band is commonly accepted as one reasonable way to characterize a sound and is assumed in the literature (Dodge and Jerse 1985). Having only two tone controls such as treble and bass means that their effect is gross and therefore readily apparent. When there is a much finer degree of control such as in professional equalizers with 10 or even 16 bands rather than just 2, it becomes much more difficult to know what part of the sound should be changed to give the desired effect.

There are two main uses of sound equalization. First, it is used to make a sound be perceived as more natural sounding. Audio equipment such as mixers, signal processors, amplifiers, speakers and even cables have the unfortunate side effect of “coloring” or changing aspects of the original sound. One reason some equipment is more expensive is that it ostensibly gives a more faithful reproduction of the original. The problem is compounded by the acoustic properties of the listening environment. The size, shape, and types of surfaces in a room affect which frequencies are absorbed and which are reflected. Equalization is used to not only compensate for the equipment, but for the characteristics of the listening environment as well.

Secondly equalization is used to give a sound a new property, such as in making drums sound more resonant or making a guitar sound “crisp.” In this use of equalization there may be some undesirable aspect of the original sound which is changed. For
instance a harsh “nasal” quality to a singer's voice may be softened or eliminated through use of equalization.

Typically when a sound engineer is setting up a sound system, the system as a whole is equalized to compensate for the equipment and the listening environment. Next individual channels are equalized for the microphones on particular instruments or other sound sources. Finally, as the entire system is used, say for instance a band playing in a live setting, equalization may be used to affect the degree to which sound sources acoustically stand out or blend. Expert sound engineers are those who have developed through experience the ability to hear a sound and isolate exactly which one or several frequencies (out of 10 or 16 bands) need to be changed to give a desired effect.

Expert equalization is a difficult perceptual task due to the complex relationship between the physical sound waves hitting our ears and perception. What we think we hear and what is physically present are not always the same thing. Sounds such as a single note plucked on an acoustic guitar consist of a fundamental frequency as well as related overtones. Over time we learn the association between these to the extent that if we hear only the overtones our mind “supplies” the missing fundamental frequency which is the “note” we think we are hearing.

For example, telephones transmit only frequencies between 300 and 3100 Hz (Tanenbaum 1981). Tuning the lowest plucked string on an acoustic guitar down one note places its fundamental frequency at 294 Hz., below the threshold of frequencies transmitted over the phone. Nonetheless, this note being played over the phone will still be perceived due to the overtones being transmitted and our brain filling in the missing part. This example illustrates that physical properties of sounds such as those affected by equalization have a very complex relationship to the perception of sound as described by timbres such as brightness, darkness, and smoothness.
Equalization is also difficult due to its context-dependent nature. In trying to make a sound brighter, for instance, it is not possible to always apply the same equalization changes. Different frequencies would need to be modified for a cymbal crash than for a bass drum hit.

1.3 Expert System Goals

The context-dependent and complex nature of equalization will be demonstrated in the expert system by comparing the subjective evaluations of experts. Comparisons will be made between a set equalization, called the “linear average,” and a context-dependent equalization determined through the use of NN pattern matching. (NN will be more fully discussed in chapter 3.) Consider for example modifying equalization to increase the brightness of a sound. If the same equalization applied to all sounds always gave the desired increase in brightness, then it would not be necessary to analyze the context of each sound. We contend that it is necessary to take the context into consideration when making equalization changes. The results of an implemented system show that the nearest-neighbor context-dependent equalization is rated 68% higher than the linear average and that it is preferred 81% of the time.

1.4 General Related Work

Our CHI paradigm of using the computer as a perceptive tool is similar to implemented systems for Virtual Reality, where the interface lies between the user and the representation of an artificial environment. For instance, a person hooked up to such a system can play a virtual game of racquetball in the lab, where the computer senses hand movements, providing visual feedback of a simulated ball through a set of miniature head-mounted display stereo graphics screens (HDTV 1990). Our work differs in that our equivalent task in this scenario would be to give the interface the facility to
respond to requests such as “make my opponent a more energetic player;” or “make them stronger.”

Visual, aural and tactile feedback are used by biochemists in the pharmaceutical drug design process through a simulation representing the atomic interaction between molecules (www.ncsa.uiuc.edu). Users get tactile feedback as they manipulate the image of a molecule they are building. In this case the computer is used as a sensory tool in the virtual environment, however it isn’t an intelligent tool in that it doesn’t learn.

There are also successes in perceptual task microworlds using different senses. A vehicle under autonomous visual control called Navlab is able to “see” through use of video cameras, laser, a global positioning system, an inertial navigation system and sonar (Pomerleau 1990). The outputs of a neural network trained by examples of humans driving specify the extent to which the vehicle should turn left or right, which is a perceptual task (Thorpe, Kanade, and Shafer 1987). Navlab has achieved autonomous driving at 55 m.p.h. for a stretch of more than 90 miles on a highway, as well as success on different types of road in various weather conditions. Our work differs from Navlab in that we do not use neural networks and Navlab is not a computer tool since it is designed to be autonomous.

An example of a computer “hearing” is automatic speech recognition (ASR). ASR for discrete-word systems under benign conditions currently has vocabularies that range up to 40,000 words (Rudnicky, Hauptmann, and Lee 1994), with vendors that include Dragon, IBM, and Kurzweil. These systems can be used to take dictation and also as a short-cut to computer commands. The dictation systems use statistical language models to favor more frequent words and word sequences. In the case of using ASR as a shortcut to computer commands, the user is an integral part of the system.

---

1World Wide Web (WWW) addresses are supplied as references for multi-media material. Full addresses are in the bibliography. A browser such as Netscape may be used to view this material.
using the computer as a tool for a perceptual task. Speech recognition is not a particularly “expert” task however, since it is shared among experts and non-experts alike.

Lee Spector’s GenBebop program (Spector 1995) responds to a four-bar piece of music input presented as MIDI signals with four bars of improvisation (hamp.hampshire.edu). The system uses a genetic algorithm to create “original” music. GenBebop learns insofar as its behavior can change depending on the underlying set of classifiers and their respective strengths. Although improvisation is an expert skill, GenBebop “performed acceptably but not spectacularly” (Spector 1995). Our work differs from GenBebop in that we perform in real-time and use nearest neighbor rather than genetic classifiers or neural networks.

Perceptual tasks are generally difficult because they involve a rich knowledge content, a high data rate and real-time response (Reddy 1988). Perceptual tasks involving aesthetics, such as making a sound more pleasant by manipulating equalization or creating a painting, may be difficult to objectively measure. People’s hearing, for instance, is different due to both physiological differences in the ear as well as varying degrees of the brain being able to differentiate and classify stimuli (ear training). In terms of the visual perceptual task of painting, consider Harold Cohen’s AARON computer system (McCorduck 1991)(Cohen H. 1995)(www.scinetphotos), which automatically generates paintings through the use of an elaborate rule-based system and a flatbed plotter. How does one measure the degree of creativity or artistic value in a painting? In such subjective areas a system can be evaluated by comparing it to what a person would do in the same situation. Another way of putting this is to use “aesthetic critical criteria as parameters” (Spector 1995).

It may also occur that even if it is possible to measure a perceptual task, we don’t know what aspect of the sensory information to measure. (Spector 1995) describes a system that attempts to inductively recapitulate deep musical structure.
Inductive learning using a neural network is used to try to capture significant aspects of the sensory information in spite of not understanding specifically what the significant aspects are.

This idea of inductive learning is one of the attractive features of neural networks (Rumelhart and McClelland 1988) (Kosko 1992). Improvement in memory and processor technology have made possible neural network advances that had previously been dismissed as theoretically impractical. For instance, the convergence theorem (Rosenblatt 1962), showed that one-layer perceptrons could learn anything they were capable of representing, but (Minsky 1975) showed the limitations of these one-layer networks. In multi-layer networks these limitations have been overcome, however the convergence theorem no longer holds. (Rich and Knight 1991) states that “the lack of a convergence theorem is not a problem in practice... this was not discovered until recently, when digital computers became fast enough to support large-scale simulations of neural networks.” Scalability in neural networks, particularly in the time it takes to train the system, continues to be a problem (Nadler and Smith1993). As (Rumelhart and McClelland 1988) states, the “combinatorial explosion catches you sooner or later, although sometimes in different ways in parallel than in serial.” Pattern matching using nearest neighbor shares this shortcoming, with the difference that even though it is not scalable, it is big enough to solve difficult tasks, given an efficient data representation and state-of-the art memory speeds and sizes.

We have seen related instances of microworlds for perceptual tasks, though none use the computer as a tool to aid the user in both perceiving and performing a task to learn an expert skill in the same way as found in this work.
CHAPTER 2

USING THE COMPUTER AS A TOOL

In this work we use the computer as a perceptual tool, where the human is an integral system component. Linking the human element in with the computer gives us the best of both worlds: the human ability to deal with the vicissitudes of the real world along with technical assistance from the computer in a few specific areas. Humans have the advantage of being adaptable and able to react to unforeseen circumstances, which computers can’t do for the most part.

2.1 Different Kinds of Tools

The computer can be used as a tool in different ways. An expert knowledge system, such as MYCIN (Shortliffe 1976), could take a file of patient lab results and one by one analyze them to give the probability of certain infectious diseases. An expert skill system, on the other hand, uses the computer as an intelligent tool to execute the perceptual task chosen by the user. For the MYCIN scenario above, this would mean the computer would assist in something like interpreting the lab test slides.

Consider again the computer artist creation AARON (Cohen H. 1995), which could be considered a computer tool to create art. In terms of the way we conceive of using the computer as a tool, we would imagine that the user would act as the art instructor, giving directives to the system such as “make that figure more humanoid,” or “give more definition to the background.” The user is required to interact with the computer, possibly for several iterations until the task is accomplished.
For the task of sound equalization, a user could request that a sound be made brighter or darker, depending on the users perception of the original sound and the goal desired. It would not make sense for the computer to go through a list of sounds, making each one brighter, for instance, since some sounds may already be bright enough, or indeed be too bright. We see then that the user is an integral part of the system, where the system’s success can be measured by its ability to accomplish the perceptual task according to the goal specified by the user.

2.2 Schema for Capturing Expertise

Consider a sound equalization task as experienced by an expert sound engineer, represented in Figure 2. Our goal is to externalize a sound engineer's internal expertise, capturing it in a form which can be reused by a non-expert. The engineer first recognizes the features or context of the present sound, then remembers similar sounds and...
equalization changes made in the past with respect to the desired outcome. This information is used to infer similar equalization changes to be made in the present case. Our intent is that this process be externalized to the point that a user can think only about the goals and need not have the expertise to match features or infer equalization changes. The same schematic would apply to perceptual tasks other than our example of sound engineering.

Figure 3 illustrates how the above schematic can be implemented in a computer system. The system must first be trained, accumulating the body of experience which constitutes the system’s expertise. The second phase, performance, uses the accumulated knowledge using inductive inference as shown by the thick light-grey lines. In order to train the system an expert user perceives the stimulus and is given a goal. The user manipulates the stimulus using the computer to achieve an aesthetically pleasing
difference with respect to the goal. The context of the original stimulus (the auditory
identifying signature), along with the new computer changes for the selected goal are
then stored in the database.

In our application to equalization the Stimulus is a sound. The Context Analysis
yields the signature of average energy per band over the duration of the sound. The
Goals are changes in the timbres of brightness, darkness, and smoothness. Each exem-
plar’s Changes are the equalizer settings used to implement the goal, and the Modifications Control is an audio equalizer.

For the performance phase, we add the inferencing module. As before, a stimu-
lus enters the system, but this time the user selects a goal. The system does pattern
matching on the stimulus’ signature, finding the $n$ most similar previously recorded
exemplars (signature-goal pairs) in the data base, using nearest-neighbor pattern matching. The system then makes the same (or very similar) changes to the present stimulus
as was made to the previously captured stimuli (nearest neighbors) for the same goal. If
the system-suggested changes are not correct, the user provides feedback, modifying the
changes. These changes are added to the database as a new exemplar. Note how the
computer is used as a tool to help perceive the input (Context Analysis), induce the
proper action to be taken (Inferencing), and also cause the resulting perceptual change
(Modifications Control). The system also has the ability to change dynamically accord-
ing to user preferences by remembering the users’ feedback in cases where the sug-
gested change was inadequate.

Now let us take a look in more detail at the Inferencing module as implemented
here using nearest neighbor pattern recognition.
Symbolic Artificial Intelligence processing is involved with “figuring out the rules,” or coming up with the underlying primitives and their relation to each other. Sometimes it is not possible to figure out the rules or in fact not necessary, particularly in cases where you are capturing a skill rather than knowledge (e.g. riding a bicycle.) In this research we are using knowledge without completely understanding its primitives and their relationships. Rather than “figuring out” or reasoning, we use pattern matching to inductively solve new problems in analogous ways to previously seen similar situations - an “expertise oracle,” as it were. Stated another way, “Intelligence is as intelligence does.” What remains to be seen is how this inductive approach can be implemented algorithmically.

3.1 Inductive Methods Background

The resurgence of feasible exemplar-based inductive learning is seen in classifier systems such as used by (Holland 1986), where he presents a classifier system using genetic algorithms to learn binary rule values and bucket brigade credit apportionment. (Goldberg 1988) discusses genetic algorithms at length, with further development on the use of induction given in (Holland 1986). Classifier systems using integer rather than a binary representation of rule values is presented in (Frey and Slate 1990), where they also develop instance-based rather than genetic rule creation and a form of credit apportionment that is more direct than that presented by (Holland 1986). (Reed 1987)
applies some of these techniques to an Othello mid-game position analyzer. (Frey and Slate 1990) also develop the ideas of fuzzy rule matching and rule strength by “accuracy” and “utility.” They compare the Classifier approach with (Quinlan 1986)’s Decision-Tree Method, as well as a NN approach (Stanfill and Waltz 1986). An efficient data representation technique using k-d trees for the NN problem is presented in (Friedman, Bentley, and Finkel 1977), with a collection of papers on k-nearest neighbors given in (Dasarathy 1991). Research by (Frey 1990) suggests NN techniques used in conjunction with Decision Trees can outperform Neural Networks, with the added advantage of greater system transparency.

Of the three methods adaptive classifiers, decision trees and NN, the third approach of NN is best suited to our task. Classifiers share the neural network drawback of a lengthy training time as well as a great deal of sensitivity to the statistics used in reward and punishment values. Decision trees allow efficient lookup but suffer from a set-up time $O(n \log n)$, as compared to $O(n)$ for NN. It is also possible when using a decision tree to exclude a neighbor that is “close” from consideration since it is in another part of the tree. The number of exemplars used in decision trees also has a strong effect on their performance. Since the maximum number of variables which can interact together in a node is equal to the tree depth, or $\lceil \log_2 n \rceil + 1$, having small datasets means that the high-order interactions occurring at lower levels in the tree are lost. Using datasets with hundreds of thousands of records solves this, but introduces the need for an adequate user interface to allow the user to visualize the tree created from the information. Although knowledge of the relationships between exemplars is more opaque when using NN, it has the advantages of being very straight-forward, sensitive to local populations, and adaptable to dynamic changes in the data.
3.2 Data Representation Issues

The concern of memory limitations can be avoided by representing the data in each exemplar in as efficient a manner as possible. We will be using an integer-vector form containing a signature of the sound (energy per frequency band), the goal, and the corresponding equalization change made to obtain the goal.

Pre-processing is important, since “a learning system cannot be expected to learn high-level concepts by starting without any knowledge at all. (Cohen and Feigenbaum1982)” We want to extract as much of the large quantity of sensory data into as compact a representation as possible. This step must unfortunately be done by hand at the present time. For instance, in machine vision a digitized video image is analyzed to detect “edges,” but a person determined that there was something called an “edge” which was important.

A related problem is selecting the level of granularity of the features which is the most relevant to the system’s goals. For instance a physicist may be most interested in the sound pressure levels in music, but a sound engineer may be most concerned about the relationship between musical tones. Choosing information at the right level for the exemplar features can make the difference between a system working well and not working at all.

The idea of doing extensive pre-processing of data before “reasoning” is done with it is exemplified in human vision. The eye preprocesses information from its 157 million rods and receptors in the retina into only 1 million optical nerve fibers connecting the eye to the brain. (Brown and Hernstein 1975)

Feature selection and scaling are also critical. Even though we don’t know specifically which features used together in some way will give the best predictive ability vis-a-vis the outcome, we must make sure they are included in the set chosen. The cor-
rect features must not only be chosen, but must also be meaningfully scaled. For instance, in predicting credit-worthiness would a marital status of *single* be better, worse, or the same as a marital status of *divorced*? The need for good feature selection and scaling means the system must be hand tuned to each application domain. This is not at all unusual for applied artificial intelligence systems.

### 3.3 Nearest Neighbor

Nearest neighbor is an exemplar-based pattern recognition approach where all the data points are stored in an n-dimensional space (hypersphere). A new example is mapped into that space and its predicted outcome is computed from the outcomes of its neighbors, that is the points close to it. These points share similar characteristics.

Consider applying NN to credit-risk analysis, where information from a credit card application is evaluated in order to assign a credit-rating to an applicant. Applicants whose credit rating falls below some threshold will not be given a credit card due to the risk involved. This is illustrated in **Figure 4**, where we are trying to predict the credit-worthiness of a loan applicant based on marital status, income, and age. The outcome, credit-worthiness, is represented by the numbers inside the boxes, and the location of the boxes themselves reflect the other fields’ values. We have chosen credit-worthiness to be scaled between 1 and 10 for this example, where 10 is most credit-worthy. Placing the new exemplar into the hypersphere of only 3 dimensions in this case we find that it is closest to two other points whose outcomes are respectively 4 and 6. The new exemplar’s outcome is then some function of those values, either by some sort of weighted average (“5” in this example) or the value that occurs most frequently in cases when there are multiple “close” values.
In our implementation of sound equalization each field (or dimension) is actually a measure of energy in one of the frequency bands averaged over the length of the entire sound.

3.3.1 Algorithm Steps and Reasoning

When it comes time to classify a new exemplar using NN, first we place all existing points in the n-dimensional space (hypersphere). This is $O(n)$ for time and memory, since we could simply read from a flat file into a multi-dimensional array. Then we find the set of nearest neighbors which are closest to the new exemplar within a maximum threshold, calling this the neighbors-set. The outcome for the new exemplar is found by taking the mean outcome of the elements of neighbors-set. This algorithm is shown in more detail below in Figure 5. The number of elements and average closeness of neighbors-set can be used as a reliability measure for the new exemplar’s outcome.
/* Algorithm to find estimated outcome value for new_exemplar. */
* Distance measure is Euclidian (e.g. ||e - new_exemplar|| is
* distance between vector e and vector new_exemplar).  e(1..d)
* means consider only elements 1 to d of vector e.  The notation
* | Neighbors_set | stands for the ordinal number of
* Neighbors_set, which is stored using a max-heap data structure.
*/

Close_threshold =
    maximum distance a point can be from new_exemplar and still vote;
Max_size_neighbors_set = number of closest neighbors to consider;
/* Initialize set of new_exemplar's m closest neighbors */
Neighbors_set = { };

/* For every stored exemplar */
FOR e = 1 TO Number_of_exemplars {
    /* For every dimension */
    FOR d = 1 TO Number_of_dimensions {
        IF ( ||e(1..d) - new_exemplar(1..d)|| > Close_threshold) THEN
            /* Get next exemplar, this one already too far away */
            break;
        } /* End FOR d = 1... */

    /* Distance from this exemplar e to new_exemplar < Close_threshold */
    IF ( |Neighbors_set| = Max_size_neighbors_set ) THEN {
        /* Neighbors set is "full".  Compare e's distance to largest
         * distance in Neighbors_set */
        IF ( ||e - new_exemplar|| < Top Heap Element THEN {
            Replace Top Heap Element with e, recreating the max-heap;
            Close_threshold = || new Top Heap Element - new_exemplar ||;
        } /* End IF ( ||e - new... */
    } /* End IF ( |Neighbors ...*/
    ELSE {
        /* Neighbors_set not full. Add e to it. */
        Neighbors_set = Neighbors_set U e;
    } /* End ELSE ... */
} /* End FOR e = 1 TO... */

Figure 5: Nearest Neighbor Algorithm
3.3.2 Domain-dependent Tuning Parameters

Fine tuning certain parameters according to the application domain can make a significant difference in the reliability of the outcome. The NN approach uses the idea of the new exemplar’s distance from existing exemplars. Different distance measurements give different results for the same problem. Some of the most common distance measurements are actually variations of the *Minkowski distance metric* (Nadler and Smith1993) given by:

\[
D_M[k, l] = \left[ \sum_{i=1}^{d} |k_i - l_i|^s \right]^{1/s}
\]

where the distance from \( k \) to \( l \) in \( d \) dimensions is city-block distance when \( s=1 \) and Euclidean distance when \( s=2 \). The distance from the new exemplar can also be used as a weight factor in determining the influence of each neighbor in choosing the new exemplar’s outcome. Again, this weight factor may not merely be linear; we may want to give closer neighbors a disproportionate amount of influence.

The accuracy of the system is also affected by the number of neighbors considered in voting, as well as the distance threshold for a neighbor to be considered. These two parameters are related to the average density of the space and the reliability of each exemplar. The data representation affects the look-up speed in real-time scenarios, such as pre-qualifying a credit applicant over the phone. Response time for a real-time system depends on the number of exemplars, the number of fields per exemplar (number of dimensions) and the required minimum proximity of nearest neighbors desired.

In our implementation we are choosing an unweighted mean of the two nearest neighbors where the closeness of neighbors is determined using city-block distance.
3.3.3 Variations to Improve Performance

Simply storing the $d$-dimensional hypersphere in a $d$-dimensional array allows us to find the nearest neighbors in $O(n \log n)$. Although not a problem when using a database with only several thousand records such as in the present case for sound equalization, this could be considered too slow if used in a real-time system. An enhancement in (Friedman, Bentley, and Finkel 1977) describes organizing the n-space using $k$-d trees, which finds nearest neighbors in only $O(\log n)$. A k-d tree can be used to find the n-nearest neighbors for a novel exemplar (a query). The k-d tree is a recursive partitioning of the entire set of exemplars by means of a binary tree, where each non-terminal node in the tree partitions its members using the mean value of some feature. The feature to be used at each node is the one with the greatest spread in values. The geometric domain of each node consists of the limits for each feature in the nodes on the path leading from the present node to the root of the tree as illustrated in Figure 6. The tree is recursively traversed by following the path which matches the query (the novel exem-

![Figure 6: $k$-$d$ Tree](image-url)
plar), examining all of the records in leaf nodes. At each point the n-closest neighbors found so far are stored in an ordered list. It is necessary to consider the records on the side of the partition opposite the current node (backtrack) only if the geometric boundaries delimiting those records overlap the ball centered at the query record with radius equal to the dissimilarity to the $m$th closest record so far encountered (Friedman, Bentley, and Finkel 1977). The algorithm stops when a “ball” defined by the query at center with radius equal to the most distant member of the n-closest neighbors is completely contained by the geometric domain of some node. For exemplars with $k$ features, this $O(\log n)$ retrieval algorithm requires file organization computationally proportional to $O(k n \log n)$.

Retrieval time is sped up by the up-front cost of file organization. There is a human analogue to doing computation at storage rather than retrieval time: it takes about 7 seconds per chunk to store a fact in the human brain long-term memory network, but retrieval can occur in only 70 milliseconds (Harmon and King 1985).

### 3.3.4 Advantages and Disadvantages of Nearest Neighbor

From a psychological standpoint, NN is close to how people think. The human associative memory model described in (Houston 1981) involves storing the deep structure or the abstract idea behind words in what is called a proposition. For instance the words “The mesmerizing sound of the surf washed over him as he left behind a trail of footprints in the sand” can be represented in a tree-like structure with structural elements of context (location, time) and fact (subject, predicate, relation, object). A partial match can return a chunk or related stored memories, for instance the smell of cookies acting as a reminder of sitting in the kitchen after school in the fall as a child. The computational analogue is an associative memory where a key is used to simultaneously match against stored patterns. This is also known as a content addressable memory,
which is used in computer memory caching and operating system page look-up tables. NN shares the idea of using a partial or full key to match against stored examples.

NN also compares favorably to the more widely used statistical regression approach to data analysis. The disadvantage to regression is that it over-generalizes when there are tens of thousands of data points. Small pockets of data which are exceptions to the general trend will be missed using regression, whereas NN is sensitive to such outliers. In fact, the more data, the better NN performs.

The fact that the application domain dictates variance in the data (e.g. “soft” social science data vs. “hard” equipment sensor data) determines how many points are needed for a desired reliability level, the same as with decision trees. “Softer” data needs larger populations for the same reliability. NN can be set up to let the user know when it “doesn’t know.” The extent to which NN “knows” is proportional to the density around the example in question.

Another strength of the system is that it can adapt dynamically using feedback from users. If some pre-set database does not fit a user’s preferences, then the feedback from the user will serve to tailor the system to the users preferences. The speed of change in the system is influenced by the size of the database and the data distribution. If the data is evenly distributed, then the speed of change in the system is inversely proportional to the size of the database. If the user makes modifications to system recommendations for examples which do not currently have close neighbors (a sparse population area), then the system will change quickly in those cases. It would change slowly in cases where there are already many similar exemplars (a dense population area).

Though not directly a problem in our implementation, there are situations where NN drawbacks can be overcome by using decision trees and NN together complementa-
rily. (Breiman, Friedman, Olshen, and Stone 1984) lists the following drawbacks to NN as justification of the need for decision trees:

1. There is no natural or simple way to handle categorical variables and missing data.

2. They are computationally expensive as classifiers; all the exemplars must be stored, the interpoint distances and distance from the new exemplar recomputed for each new exemplar.

3. They are sensitive to the choice of the distance metric, and there is usually no intrinsically preferred definition.

4. They give very little usable information regarding the structure of the data.

Though not a problem with our numerical attributes of energy per frequency band, (Frey 1990) has addressed the concern given in point 1 above by transforming categorical features (such as marital status) into numerical ranges by using a decision tree. For instance, in Figure 4 we saw the features of income, age, and marital status used to classify credit-worthiness. Income lends itself easily to the distance measures used to find the nearest neighbors, but how can a linear ordering be imposed on the categories of marital status? In terms of the outcome of credit-worthiness, should “single” or “married” be closest to “divorced”? The problem is not only what should the ordering be, but how “far apart” should these categories be on the marital status dimension relative to the credit-worthiness outcome? To solve this problem a decision tree is built using only the categorical attribute in question as a splitting variable (marital status in this case.) Each exemplar is then given the mean outcome value for the leaf node into which it falls, yielding a linear ordering (Frey 1990). We can infer from the example in Figure 4 that “single” and “divorced” are the most different as predictors of credit-worthiness, with
“married” being more similar to “divorced” than to “single.” This same approach can be used to combine several features into a composite feature. Time and memory for scaling is approximately $O(n^2)$.

The contention of NN being computationally expensive (point 2 above) is a valid concern, though it does not pose an intractable obstacle in the face of developed technologies, as mentioned on page 9. His point about the distance metric (point 3) can be overcome by experimentation in each application domain. The fourth objection, that of opaqueness of the data representation, is the most serious, though only from a symbolic processing perspective. NN can still be used successfully in domains where either we don’t need to know about the structure of the data or it is taken care of by extensive preprocessing during feature selection.

NN sounds nice in theory, but is it tractable in practice, particularly when applied to a difficult perceptual task? The next section describes our implementation using NN showing that it can indeed perform the difficult perceptual task of sound equalization.
CHAPTER 4

NEAREST NEIGHBOR INDUCTIVE INFERENCE TO CHANGE TIMBRE USING EQUALIZATION

The goal of this study was to create a trained system usable as a tool by a non-expert to do expert sound equalization (eq). Changing the quality, or timbre of a sound using equalization through an implementation of a NN inductive inference system is an example of applying an existing technology to a new area. Advances in machine learning have often come from empirical studies where knowledge representations are adapted to new domains. The notable AI researcher Herbert Simon writes: “a large part of our understanding of intelligence - artificial as well as natural - will continue to depend upon experimentation, [where] much theory in AI will be relatively qualitative and informal.(Simon 1993)” This has been true over the years as seen in examples such as checkers playing using discriminant functions (Samuel 1959), rule-based medical diagnosis by MYCIN (Shortliffe 1976), rule-based configuration of DEC computer systems with R1 (McDermott 1982), and 3-dimensional VLSI design using EURISKO (Lenat 1983).

The question which we wished to answer with this study was whether or not context needs to be taken into account in affecting timbre through equalization. In order to answer this we compared the mean change across all sounds and all users for the desired goal (linear average) against a context-dependent change as calculated using nearest neighbor pattern matching for the same goal. We found that the context must be taken into account when doing equalization and learned some lessons regarding interface design and perception of sound in the process.
The tool used in this study was an on-screen equalizer implementing *nine* different frequency band controls rather than the usual ten.² These nine controls can be thought of as dividing the sound into nine frequency bands, where each is controlled by its own equalizer slider. The lowest bands (low frequencies) were on the left, rising up to the highest bands (high frequencies) on the right. The computer interface recorded each user’s equalizer settings along with other context-identifying information.

Two experiments were run in this study. In the first, length of training time and variability in the training environment obscured the relationship between the linear average equalization and the nearest-neighbor calculated equalization, but did provide valuable insights into the interface and experimental design. In the second experiment a more constrained setup clearly showed users’ preference for the nearest-neighbor calculated context dependent equalization. We will describe the first experiment and the lessons learned and then discuss the second. First however we will discuss the selection of timbres used in both experiments.

### 4.1 Adjusting Timbres of Brightness, Darkness, and Smoothness

The task of changing timbre through sound equalization is exercised by experts constantly in the real world, particularly sound engineers modifying sound equalization for recordings or live settings. We looked specifically at the timbres of brightness, darkness, and smoothness, with loudness as a control (*Figure 7*). We ignored the temporal characteristics of sound (e.g. reverberation) and considered only the mean energy per

---

²It was originally 10 bands, but since we are sampling at 22.05 kHz, we can only represent frequencies up to 11 kHz, so the 10th band at 16 kHz was disabled. According to Nyquist’s sampling theorem, the highest frequency that can be accurately represented is half the sampling rate, which is 22 KHz. in our case (Dodge and Jerse 1985)
band over the time of the sound segment being played. From our own observation this seems to reasonably be the case for human experts as well.

The timbres of brightness, darkness, and smoothness were chosen because they are featured as principal timbres in the literature (Von Bismark 1974)(Grey 1977), and sound engineers we interviewed deemed them useful and widely accepted terms (Bartlett and Bartlett 1995). *Brightness* can be thought of as high-frequency emphasis, with weaker low frequencies. *Darkness* can be defined as low-frequency emphasis, with weaker high frequencies - essentially the opposite of brightness. Thirdly, a sound is *smooth* if it is easy on the ears, not harsh, with a flat frequency response, especially in the mid-range, with an absence of peaks and dips in the response. The control, *loudness*, is an overall increase in the energy level present in the sound.

As mentioned previously, what makes this equalization task difficult is that the equalization changes are *context dependent*. What makes one sound brighter may not work for another. Making a cymbal brighter would involve increasing the energy in the highest frequencies available (the sliders furthest to the right), but doing the same thing to an electric bass sound may not make any difference at all since there is *no* energy
present at those high frequencies to begin with. When adjusting sliders to make an
equalization change, one must take into account the characteristics of the underlying
sound. It isn’t possible to just always do the same thing to every sound for a desired
effect. Not only are the equalizations context dependent, but they are non-linear as well.
Moving certain sliders could make a sound increasingly smooth, but after a point con-
tinuing to move the same sliders in the same direction could give an unpleasant quality
to the sound.

For example, consider the goal of an increase in brightness applied to the three
sounds (bass, acoustic guitar, and rainstick) shown in Figure 8. To make the bass sound
brighter we would want to increase the energy in the bands 500, 1K, and 2K. To make
the rainstick sound brighter, however, we would have to increase the energy in bands
2K, 4K, and 8K, which is different.

One of the concerns in having the computer perform the perceptual task of
equalization is that the result could be evaluated as subjectively different between one
person and another. This subjective nature of listeners’ perception of timbres is of some
concern, but has been studied previously. (Von Bismark 1974) compared scales of oppo-
site verbal attributes used to describe the perception of sound (e.g. sharp-dull, rough-smooth, relaxed-tense, thick-thin, heavy-light, compact-scattered). He gathered 25,090 ratings over 30 such verbal timbral scales, observing that “the timbre of musical sounds and speech sounds is strongly affected by their time structure.” Consequently he controlled the experiment to “deal only with steady sounds equalized in loudness, pitch (i.e. fundamental frequency) and duration.” Of the 30 timbral scales used, von Bismarck found that there was remarkable agreement between subjects on describing timbres on the largely independent scales of dull-sharp, compact-scattered, full-empty and colorful-colorless.

Part of the motivation for Von Bismarck’s work was the sense that “similar to the [well understood] attributes of loudness and pitch, language may have coined expressions which are indicative of other major dimensions of sound perception.” Von Bismarck’s results show that not only were the timbral ratings consistent over time for each individual, but they were consistent across the entire group of subjects, with a correlation coefficient exceeding 80%. This was true for both musicians and non-musicians.3

The above studies gave us confidence that there would be meaningful agreement between subjects in the definitions of the timbres used.

4.2 The First Experiment

The first experiment had two parts to it: first the training phase, where subjects used the computer to make equalization changes with the computer remembering what they did; and second the testing phase, where users gave feedback as to how good of a job the computer did in making equalization changes.

3See also (Grey 1977) for a timbral multidimensional graphing of orchestral instruments, which shows brightness to be a primary characteristic of a group of instrumental sounds.
The interface consisted of an on-screen equalizer implemented on a 25 MHz NeXT 68040 computer. The 10-band on-screen equalizer\(^4\) had the top (10th) band at 16 kHz. disabled, as explained on page 27. We designed and wrote software consisting of approximately 5,000 lines of assembler and Objective-C code implementing the graphical user interface (GUI), DSP DMA, and real-time measurement of energy per frequency band. The computer system reproduced sounds through a Carver amplifier and a pair of Yorkville YSM-1 near-field monitors.

The sounds used were taken from unprocessed studio master tracks of typical folk/rock music, consisting mainly of vocals, different electric and acoustic guitars, basses, keyboards, flute, drums and other percussive sounds. 42 stereo sound segments approximately 15 seconds long each were used for the training session, with an attempt made to have sounds representative of different frequency energy distributions. The testing session sounds were a distinct set of 10 more sounds. The sound segments were stored in a 16-bit digital format at a 22.05 kHz. sampling rate. A signature, or measurement of root-mean-squared (RMS) energy per each of the nine frequency bands over all 15 seconds was taken for each sound, with a filter at 40 dB\(^5\) to exclude quiet spots in the sound segment in the averaging. For example, we did not want the measurement of average energy in a drum sound to include the silences between beats. The signature of energy in the nine bands was used to place each sound in a nine-dimensional space (nine dimensional array) for searching using nearest neighbor. It has been previously mentioned that increased processor speeds and size of memory have made previously intractable problems now feasible (page 1). The difficulty of the present problem of sound

---

\(^4\)The equalizer is a 10-band (octave) Infinite Impulse Response (IIR) filter with a band-pass width, or quality factor (Q) of 1.4, a gain of +/- ~ 14 dB, with bands centered at 31, 62, 125, 250, 500, 1K, 2K, 4K, and 8K Hz.

\(^5\)dB here is used as a measurement of acoustic intensity (Dodge and Jerse 1985), where conversational speech has an intensity of approximately 60 dB. Each doubling of the amplitude of the sound corresponds to a 6 dB increase.
equalization is due to the need to analyze sensory data in real-time (RMS in our case), rather than to concerns over database sizes. An early implementation of this system on a Motorola 68030 (rather than 68040) CPU was in fact too slow to use in real-time.

The 11 subjects were a mixture of musicians and sound engineers, where 9 of the 11 are professionals. Because of the difficulty in scheduling training sessions, we decided to take the computer system, amplifier and speakers to the subjects’ work place or home. This meant that the listening environment was different in each case. Since the system was usually left set up overnight for the subjects to use and the experimental session were not monitored, some subjects took breaks while others went straight through the session. These factors caused problems later.

4.2.1 Training Phase

Each subject spent between 2 and 4 hours on the training phase using the interface shown in Figure 9. To start the first sound playing users would select the “Play” button. The desired goal was highlighted in the goals window (i.e. more or less of Brightness, Smoothness, or Loudness) so users would adjust the sliders to make a “just noticeable difference” (jnd) relative to that goal (Gescheider 1976). The “Flat Eq” and “Changed Eq” buttons allowed subjects to compare the changed sounds to the original sounds. These controls were used in real time, while the sound was being played. The sound could be replayed as many times as needed. Once subjects were satisfied that the goal had been met, selecting the “Next” button took them to the next training example.

For each user for each sound-goal combination the computer created and stored an exemplar consisting of:

- Soundfile name
- Goal (one of 6 from the goals matrix)
- Final slider positions (scaled from 0..31 for each slider)
- Energy-per-band “signature” for that sound. (RMS)
The equalizations for all six goals for each sound were completed before advancing to the next sound.

4.2.2 Testing Phase

The same 11 subjects used to train the system continued with the testing phase. The accumulated data for the 11 selected subjects became the dataset of 462 examples (11 users x 42 sounds) per each of the six goals, for a total of 2,772 exemplars.

Subjects were asked to give a rating to each of 4 equalizations relative to the highlighted goal. These four types of equalizations are shown in Figure 10, where each consisted of the mean equalization across some set of exemplars. The $y$ axis shows whether all or simply the 4 nearest neighbors were chosen, and the $x$ axis shows whether

![Figure 9: Training Screen for Experiment 1](image-url)
exemplars from all users or simply the subject’s exemplars were considered. This arrangement was to help answer two questions:

1. Are the equalizations context sensitive? This could be determined by comparing the top row (All Exemplars) with the bottom row (NN Exemplars). Better values in the bottom row would suggest the equalizations are context sensitive.

2. Is there general agreement as to the definition of the equalization timbral terms brightness, smoothness, and loudness? This could be determined by comparing the left column (All Users) with the right column (One User). Little difference between the two columns would suggest agreement on these definitions.

The four equalizations were applied in turn by selecting buttons in the interface.

The interface screen for the testing phase is shown in Figure 11. Each of the four equalizations was represented by one of the eq selection buttons, with the rating given by moving the matching slider directly beneath it. The correspondence between the four equalizations and the eq selection button positions were randomized on each presentation.

To start the sound playing, subjects selected the “Play” button. While the sound was playing one of the four equalizations could be applied by selecting “Eq A”, “Eq B”, “Eq C” or “Eq D.” This selected equalization could be compared to the original un-equalized sound by clicking on the “Flat Eq” button. Users went back and forth between

![Figure 10: Four Equalizations](image-url)
these two buttons (one of the buttons A, B, C or D and the “Flat Eq” button) until they were satisfied as to how good of a job this equalization (A, B, C or D) did relative to the goal highlighted in the goals window. A rating was then given to this equalization using the corresponding eq rating slider beneath the selected eq selection button. These steps were repeated for the other three eq selection buttons (the three not yet selected from A, B, C or D). Once the user was satisfied with the ratings given to the four equalization options clicking on the “Next” button advanced to the next example.

4.2.3 Results

The experiment indicated that equalizations are indeed context sensitive and that there is general agreement on the definition of the equalization timbral terms brightness, smoothness, and loudness. These results were not conclusive, however, due to the unacceptably high standard deviation (SD). The means of the 4 types of equalization ratings

![Figure 11: Testing Screen for Experiment 1](image-url)
are shown in Figure 12. The bottom NN row values are an average 0.5, or 7% higher than the top “All Exemplars” row, supporting the hypothesis that equalizations are context dependent. The left column (All Users) and right column (One User) differ by only an average of 0.085, or 0.6%, supporting the hypothesis that there is general agreement on the definition of the timbral terms. The problem is that the standard deviation is 4.45, which overlaps the differences upon which our conclusions are based.

When giving equalization ratings, subjects would sometimes leave a rating slider in the bottom-most “unchanged” position in cases where they could not hear any discernible difference. In the process of analyzing the data it became evident that there was a vast difference between subjects’ ability to hear some of the equalizations, with the number of unrated equalizations ranging from a low of 6% to a high of 50%. In other words, in the worst case the subject could not discern the equalization half of the time. This large number of unrated cases contributed to the large standard deviation.

The large number of unrated cases was addressed by only considering those exemplars rated by at least \( x \)% of the subjects, where \( x \) ranged from 45% to 91%. The different values for \( x \) are shown in Figure 13. This shows that a large number of cases were rated by 8, 9 and 10 (out of 11) subjects, with a smaller number of cases rated by 5, 6, and 7 subjects. There were only 8 cases rated by all 11 subjects. We considered only

![Figure 12: Equalization Means](image)

<table>
<thead>
<tr>
<th></th>
<th>All Users</th>
<th>One User</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Exemplars</td>
<td>6.86</td>
<td>6.79</td>
</tr>
<tr>
<td>NN Exemplars</td>
<td>7.27</td>
<td>7.37</td>
</tr>
<tr>
<td>SD</td>
<td>4.45</td>
<td></td>
</tr>
</tbody>
</table>
those cases rated by at least 8 out of the 11 subjects, with the resulting frequency counts per equalization type shown in Figure 14. What stands out is the difference between the “One User - NN Exemplars” count (19) and the other three counts (8 or 9) for the Smoothness goal. This indicates that subjects’ ability to hear equalization changes is strongly context dependent for the smoothness goal. Interestingly, subjects had also indicated that smoothness was the most difficult timbre to implement.

Grouping outcomes according to clusters of sound types (predominantly “bass sounds” vs. “treble sounds”) using a decision tree failed to yield any results that would
overcome the large standard deviation values. The presentation order of exemplars seemed to have a small effect, as the context-dependence trend became more noticeable toward the end of the testing session.

4.3 Experiment Modifications and Lessons Learned

Even though the resulting data from the first experiment suggested that context must be considered in doing equalization, it was not conclusive. The large variance in the data indicated by the standard deviation suggested that there were too many variables present in the experiment. Various factors were analyzed in preparation for the second experiment.

One of the main reasons for the weak results seemed to be that the acoustic environment varied between subjects. The system (computer, amplifier, speakers) was carried from location to location, sometimes being set up in subjects’ workplace, sometimes in their homes. Consequently the amount of ambient noise and the acoustics of the room varied from location to location, affecting subjects ability to hear. This was addressed by setting up a consistent listening chamber used in all sessions. This listening chamber was lined with SONEX sound baffling panels to eliminate early reflections, with physical dimensions as shown in Figure 15. Technical tests (TEF) of the Yorkville monitors revealed a frequency response dip at around 1K, possibly due to the crossover. This had the effect of not faithfully reproducing sounds in that frequency range. More technical testing was done to select a matched pair of Genelec 1030A powered near-field monitors for use in the experiment. These monitors are some of the very best available anywhere, giving extremely accurate reproduction (flat response).

A second problem was the bimodal nature of the goals. Subjects were asked to rate the goodness (1 to 15) of equalization adjustments, where the best rating should have been given to equalization adjustments making a just noticeable difference (jnd)
vis-a-vis the goal, as illustrated in Figure 16a. (The six goals were more & less of brightness, smoothness, & loudness.) This means that BOTH gross equalization changes and no change should have given a worse rating. Having two separate conditions which both were supposed to be rated as poor contributed to the variance in ratings. Some subjects apparently just rated sounds as to how “good” of a job the equalization did vis-a-vis the goal, as illustrated in Figure 16b.

In order to address this in the second experiment subjects were asked to give ratings of equalizations based on the aesthetic “goodness” of the equalization relative to the goal rather than using jnd. Also fewer goals were presented the second time. Rather
than more and less of brightness, smoothness, and loudness (6 goals), only 3 goals were presented: more of brightness, darkness and smoothness. This cut the training time in half, helping prevent the auditory fatigue of which some subjects had complained. For the same reason sounds were also edited to have a more uniform gain level, and the overall system loudness level was lowered. All the different sounds for the same goal were also now grouped together, so users did all “more brightness” changes before going on to another goal. This helped negate the effect that presentation order had.

It turned out that the sounds used in the testing phase in the first experiment had predominantly low and mid-range energy. Several changes were made to provide a more representative set of sounds, where sounds chosen for the test phase were representative across the spectrum for a rock-type multi-track recording as shown in Figure 17.

Some subjects took over twice as long for the same task as others in the first experiment. This was remedied by having the sessions more structured, with a supervisor present, a formal handout of instructions and pre and post questionnaires asking about the subject’s understanding of the goals, their ease of doing it, and their ability to hear equalization changes.
Feedback from the first experiment indicated subjects were frustrated in trying to implement goals that did not make sense for some sounds (e.g., making an already-extremely-bright cymbal “brighter”). These sound-goal pairs were eliminated for the second experiment.

Data from the Testing phase in the first experiment was difficult to analyze. Rather than present *four* equalization possibilities for rating by the subjects as shown in Figure 10, there were now only *three* equalization possibilities rated by the subjects. These three were: (1) The average equalization for all exemplars for all users, (2) The average equalization of the nearest neighbors for this subject only, and (3) No change. This essentially was the two “extremes” from Figure 10 with the “no change” rating acting as a baseline against which we compared the users’ ratings. This made it easier to test our hypothesis of needing to take context into account for equalization.

Another problem was that there was no base-line figure on the aural acuity of subjects. Besides the “no change” rating mentioned above, we also constructed a brief

![Energy per Band in Testing Phase Sounds](image)

*Figure 17: Energy per Band for Test Phase Sounds*
hearing test consisting of multiple equalizations ranging from “no change” to “grossly changed.” Subjects’ consistency in doing the equalization rating task was also measured by having two duplicate sounds during the test phase, rather than only one such duplicate the first time around.

Subjects found it tedious to constantly have to re-select the “Play” button during iterations for a single equalization goal. This was circumvented by modifying the interface to play the sound (if it was not already being played) anytime the subject moved an equalizer slider or selected an equalization button. This had the effect of keeping the sound playing as long as the subject was making modifications or comparing equalizations.

Finally we concluded that the number of nearest neighbors was still too large: Averaging too many neighbors had the effect of masking the context dependance of the task. The smaller the database, the smaller the number of nearest neighbors should be. The mean of the 4 nearest neighbors was used the first time, but only the 2 nearest neighbors were used in the second experiment. The distance basis for the number of nearest neighbors considered is shown in Appendix A on page 56.

4.4 The Second Experiment

Like the first experiment the second had a training and testing phase, with the addition of a preliminary brief hearing test. The same 42 15-second stereo sound segments were used as in the first experiment, with the exception of one sound being eliminated due to poor quality.

4.4.1 Hearing Test

Each user was given a hearing test using the interface shown in Figure 18. For each sound the user would click back and forth on the “Eq A” or “Eq B” buttons as the
sound was being played. One of these buttons represented an equalization change, while the other was the unchanged sound. Once the user was satisfied as to whether the two options made the sound different or not, the “Same” or “Different” button was selected, which also had the effect of advancing to the next example. The system kept track of how many times the user judged correctly. Three sounds were used as examples, where each sound was heard 10 times for a total of 30 judgements. The 10 instances of the same sound had varying degrees of equalization change associated with them, ranging from grossly different to minutely different to no change at all. Additionally, the presentation order was randomized. More than just a hearing test, this also tested the subjects’ abilities to interpret equalization changes. Results across all users ranged from 60% to 90% correct.
4.4.2 Training Phase

Seventeen sound reinforcement professionals and students in Northwestern’s school of music served as subjects, each spending approximately 2.5 hours on the training phase using the interface shown in Figure 19. Note that this time there was only three, rather than six goal possibilities in the goals window. To start the first sound playing, users would either select an equalization (Eq) button or move one of the sliders. (The “Play” button was disabled.) The goals were highlighted and equalization sliders used as in the first experiment. All 41 examples with “Brightness” as a goal were done first, then those for “Darkness”, and finally “Smoothness.” The detailed instructions given to the subjects for this phase are shown in Appendix B starting on page 58.
4.4.3 Testing Phase

The 17 subjects from the training phase were evaluated to select the best 11 subjects to continue with the testing phase. These 11 were determined by analyzing the extent to which they moved the sliders. Users who had to move the sliders to an extreme in order to effect a perceptible change in the sound were eliminated. As expected, it turned out the better the subject’s hearing as measured by the hearing test, the less the subject tended to move the sliders, as shown in Figure 20. The accumulated data for the 11 selected subjects became the dataset of 451 examples per each of the three goals, for a total of 1353 exemplars.

The interface screen for the testing phase is shown in Figure 21. This time users were asked to give a rating to each of three different equalizations presented by the computer. These three choices were a linear average, NN average, with no change used as a control. Each of these equalizations was represented by one of the Eq Selection Buttons. The linear average was the mean slider change across all 11 users for all 41 sounds for the current goal. At the other extreme the NN average was the mean slider change of the 2 nearest neighbors for the subject only. The nearest neighbors were computed by comparing the example’s signature (energy per each of the 9 bands) with the signatures of

![Figure 20: Degree to which subjects moved sliders](image-url)
the stored data. This was essentially placing the example point in a 9-dimensional space and finding the two closest points. The correspondence between the above three equalizations and the eq selection button positions were randomized on each presentation.

To start the sound playing, subjects selected one of the Eq selection buttons, “Eq A”, “Eq B”, or “Eq C.” (As in the training phase, the “Play” button was disabled.) This selected equalization was then compared to the original sound by clicking on the “Flat Eq” button. This process continued in the same way as that in the first experiment. The eq selection button for “no change” was actually an identical setting to the “Flat Eq” button, a fact that was not always recognized by the subjects.

4.4.4 Results

The experiment validated the hypothesis that nearest neighbor pattern matching does a better job at equalizations than does a linear average. As shown in Figure 22, the

![Figure 21: Testing Screen for Experiment 2](image-url)
mean evaluation of the linear average over all users equalization was 6.00, the mean evaluation of the 2 nearest neighbors equalization was 10.08, and the mean evaluation of the no change equalization was 2.11, with a standard deviation of 3.57. Computation of standard deviation excluded unchanged ratings. Significantly, the nearest neighbor equalization was greater than the linear average by more than the standard deviation (6.00 + 3.57 = 9.57, which is still less than 10.08).

Rank ordering (Figure 23) was used to measure the extent to which the experiment followed the desired trend. It is clear that the “Nearest Neighbor” equalization option was preferred almost all the time, the “All Users Average” (linear average) was second, and the “No Change” equalization usually came in third. The full rank order results per user are given on page 68 in Appendix C.

Looking at the rank ordering results by goal (Figure 24), we see that brightness was the easiest for subjects to identify, darkness was the next easiest, and smoothness was the hardest. For smoothness in particular we can see that the “no change” equalization actually came in first place 15 out of the 110 possible times, or around 14% of the time. This indicates that smoothness was the most difficult equalization for subjects to perform. Figure 25a shows the same thing, except broken down by user. While the “No change” control condition equalization was easily identified for Brightness and to a lesser extent Darkness, it was not always correctly identified for Smoothness. In Figure
we can see that the users who moved the sliders the most (those further to the right in the graph) also most strongly preferred their own equalization settings, though this is not a particularly strong trend. This runs counter to the intuition that the more expert users (those further to the left) would more strongly prefer their own equalizations. There are various reasons for this as reported in the subjects’ post-session questionnaires that are discussed under future work.

**Figure 23**: Rank Ordering Overall

**Figure 24**: Rank Ordering by Type of Eq, Broken down by Goal
Since users applied ratings to sounds that had duplicates, we were able to measure the consistency of their ratings. Surprisingly, there was no significant correlation between the consistency of their ratings and their tendency toward the desired trend of preferring NN eq’s. Neither was there a correlation to acuity (Hearing test) or to variability of slider movement. In Figure 26 we can see that there is a slight correlation indicating that subjects who tended to be more consistent (trend line further down) also tended to do a better job at correctly identifying the “No Change” equalization choice (triangles higher up). The full variability results are shown on page 70 in Appendix C.

**Figure 25:** Eq Variability by User

**Figure 26:** Inconsistency Results
To summarize the results from the second experiment, the average rating given to NN equalizations were 68% higher than the linear average. Rank ordering showed that NN equalizations were preferred 81% of the time. Brightness changes were the easiest, then darkness changes, with smoothness changes being the most difficult for subjects to perceive and rate. The system was successful in using NN to perform the perceptual task of equalization.
CHAPTER 5

CONCLUSIONS AND FUTURE WORK

The implementation of NN to do sound equalization was successful in learning to perform a perceptual skill. This thesis shows that combining fast processors and large memories make a pattern-recognition approach feasible for perceptual tasks, where the computer is used as a perceptual tool.

One possible objection to this approach is that in implementing this as a dynamic system able to adapt to a particular user, as mentioned on page 15, the underlying database size is always strictly increasing: it doesn’t forget. We contend that for the most part, memory limitations is not an issue here. With data preprocessing and efficient data representation (such as 1 byte per attribute, or equalizer band in our case), millions of exemplars can be stored. Our exemplars could be represented using only 21 bytes each. Current architectures of desk-top computers permit expanding up to 256M of main memory, which would give us a database size on the order of 12 million exemplars. Assuming 15 seconds per sound segment, it would take ~5 years of constant use, 24 hours a day, to accumulate this many exemplars on a single system. If absolutely necessary, such as a case where a database were constantly changing in real-time from multiple sources, a first-in-first-out (FIFO) or least-frequently-used (LFU) exemplar replacement strategy could be used.
5.1 Present Practical Applications

Not only does the theoretical idea work, but it has practical applications as well. It could be implemented with a preset hard-wired database as a consumer “black box” to do expert equalization. Current home stereo systems have equalization presets for types of music such as “classical” and “jazz.” The problem with these is that the equalization should be different depending on what kind of jazz, for instance, is being played. A system such as that shown in the present research would allow for adaptive context dependent equalization. As the underlying type of music changed, the equalization would correspondingly change.

Changing the timbre of a sound through equalization is a common task exercised by audio engineers in the real world in making recordings or live sound mixing. Engineers could train the system with their favorite settings. Figure 27 shows the screenshot of a working prototype of this idea. It implements expert equalization using the data gathered in the training phase of the experiment, where the equalizations that the computer suggests are context dependent and are customized to user preferences. It is also possible to adjust the extent of the chosen effect. For example, in the screen shown here the “Darkness” button has been selected, which changes the equalization sliders to give that effect. Moving the slider under the “Darkness” button upwards makes the sound even more dark, while moving it downwards makes it less dark. This is done by first using NN to see which sliders should change and in what proportion to each other. Choosing more and less of the desired effect changes the extent to which the sliders move while maintaining their relative proportions.

5.2 Future Work

The implementation using NN worked well, but it would be interesting to try other methods such as a decision tree by itself. One way to look at the system is that it
implements a many-to-one mapping, putting many complicated controls into a single control that appropriately affects the outcome.

The paradigm presented here could also be used to exploit the computer as a tool in extending users' perception in the modalities of sight or smell or other applications in hearing. Using nearest neighbor for a perceptive task could be used by airlines in interpreting video or x-ray data in explosives detection in luggage (Murphy 1989) or by the Navy in interpreting audio signals for submarine detection.

Post-experiment feedback from the subjects indicated their frustration with the chosen timbres of Brightness, Darkness, and Smoothness. They expressed a desire to simply make each example sound “better.” It would be interesting to repeat the experiment using “better” as the timbral goal. It was also apparent in the experiment that users used different criteria when making equalization changes in the training phase than they

![Figure 27: Working Prototype of an Expert Equalizer](image)
did when *recognizing* equalization changes in the testing phase. This could be overcome by having users choose between many preset equalizations in the training phase rather than have them move the sliders.

There were instances where NN clearly did the best job of making a sound “brighter,” however the increase in gain also made it sound unpleasant so it was not the subject’s first choice. Getting a change in timbre without a change in gain is an example of composite goals and the interdependence of goals (Cohen and Feigenbaum1982). Solving this problem would be very valuable but difficult to achieve, since different people have varying sensitivity to different frequencies, so the comparative gain between two sounds might be the same for one person but not for another.

The most interesting avenue for exploration with the type of user interface presented here would be to include temporal information, with the desired effect being control over a sound quality such as reverberation. Complex relationships between many controls such as filters and delays could be given a single control to help localize sound in 3 dimensions (Kendall and Martens1984), determine physical characteristics of the virtual listening environment, and set the perceiver’s location in the space.

In conclusion, we have seen how modelling an expert skill is different from modelling expert knowledge. Inductive learning is an appropriate strategy to capture human expertise in domains with complex relationships in the underlying data which would not be possible to model in a traditional rule-based system. Advances in computer processor and memory technology have brought perceptual tasks within the fringes of what can be done using pattern recognition techniques. The nearest neighbor inductive learning technique can successfully be used to develop an assistant in performing the perceptual task of equalization adjustment.