2012-08-13

Automatic Generation of Music for Inducing Emotive and Physiological Responses

Kristine Perry Monteith
Brigham Young University - Provo

Automatic Generation of Music for Inducing Emotive and Physiological Responses

Kristine P. Monteith

A dissertation submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Tony Martinez, Chair
Christophe Giraud-Carrier
Dan Ventura
Mike Jones
Bruce Brown

Department of Computer Science

Brigham Young University

December 2012

ABSTRACT

Automatic Generation of Music for Inducing Emotive and Physiological Responses

Kristine P. Monteith
Department of Computer Science, BYU
Doctor of Philosophy

Music and emotion are two realms traditionally considered to be unique to human intelligence. This dissertation focuses on furthering artificial intelligence research, specifically in the area of computational creativity, by investigating methods of composing music that elicits desired emotional and physiological responses. It includes the following:

- An algorithm for generating original musical selections that effectively elicit targeted emotional and physiological responses,
- A description of some of the musical features that contribute to the conveyance of a given emotion or the elicitation of a given physiological response, and
- An account of how this algorithm can be used effectively in two different situations: the generation of soundtracks for fairy tales and the generation of melodic accompaniments for lyrics.

This dissertation also presents research on more general machine learning topics. These include:

- A method of combining output from base classifiers in an ensemble that improves accuracy over a number of different baseline strategies, and
- A description of some of the problems inherent in the Bayesian model averaging strategy and a novel algorithm for improving it.

Keywords: automatic music generation, computational creativity, ensembles, Bayesian model combination

ACKNOWLEDGMENTS

> ***Any composer who is gloriously conscious that he is a composer must believe that he receives his inspiration from a source higher than himself.* –John Philip Sousa**

I am grateful for the many contributions of my advisor, Tony Martinez, in the course of my research. In addition to his invaluable support and helpful suggestions, his extraordinary patience is greatly appreciated by me and by my family.

I would also like to acknowledge my committee members for providing wonderful suggestions and being a delight to work with. I would particularly like to thank Dan Ventura, who provided considerable support in helping me refine my topic and co-authored five papers in this work, and Bruce Brown, whose assistance was invaluable in helping me set up the biofeedback experiments and co-authoring the paper describing them. Without their help, this dissertation would probably not have been completed.

I would like to thank James Carroll, Neil Toronto, and Kevin Seppi, co-authors on one of the papers in this work, both for their help on my research and for the stimulating ideas they shared with me, as well as all the faculty, staff members, and fellow students in the CS department whose friendship has been one of my favorite parts of this entire experience.

I appreciate the musical contributions of Heather Hogue and Paul McFate, both of whom composed for this work. I would also like to thank all the research subjects who participated in my experiments. I would particularly like to thank my parents, siblings, and siblings-in-law for not only participating as research subjects–often at the last minute–but also for their prayers, phone calls, well wishes, and help with babysitting so that I could finish this dissertation.

I would like to thank my wonderful husband, Adam, for more things than I could possibly list here. Marrying the love of my life seems to have made everything in my life better, and that has extended to my research and graduation efforts as well. I would also like

to thank my daughter, Anna, for helping by not destroying my computer and being patient through numerous hours of babysitting while Mommy was busy trying to graduate.

Finally, I would like to express my gratitude to my Heavenly Father for providing me both this opportunity and the capability and grace necessary to finish it successfully. The move to BYU and the PhD in Computer Science were His idea in the first place, and I am so thankful for that inspiration.

Soli Deo Gloria

Table of Contents

Part I

Introduction

"Computing is not about computers any more. It is about living."
–Nicholas Negroponte

This dissertation deals primarily with the challenge of automatically generating music that elicits particular emotional and physiological responses. It also includes some research on more general topics in the area of machine learning. The document is divided into four sections. Part I provides an introduction to the topics covered. Part II presents papers that specifically pertain to the area of music generation. Part III presents papers on more general machine learning topics. Part IV summarizes the contributions of this research and outlines possibilities for future work.

Chapter 1

Background, Motivation, and Overview

"Ours is a history of self-imitation." **–Pamela McCorduck**

One of the main goals in the area of artificial intelligence is to design algorithms that will allow computers to behave more like humans. In some cases, the goal is purely theoretical. As one author explains "Looked at in one way, ours is a history of self-imitation...We are ten times more fascinated by clockwork imitations than by real human beings performing the same task." From cave paintings and sculpture to Greek myths and more modern tales of Frankenstein's monster or *I, Robot*, history is full of examples of mankind's tendency to create things in its own image. The study of artificial intelligence could simply be seen as the latest step in that process. But in many cases, helping machines behave in a more human-like manner can also be very practical. From speech and handwriting recognition to medical diagnoses, navigation, and fraud detection, computers can now perform a wide array of tasks once left to humans [Mitchell, 1997].

More recently, researchers have been turning their attention to more creative endeavors. Computers are being "taught" how to carry on conversations [Weizenbaum, 1966, Saygin et al., 2000], generate works of art [Norton et al., 2010], and compose text for poems and stories [Gervás, 2001, Riedl, 2004]. One major area of human creativity involves the production of music, so naturally, many computer science researchers have turned their attention to musical computation tasks. Researchers have attempted to classify music, measure musical similarity, and predict the musical preferences of users [Chai and Vercoe, 2001, Li and Ogihara, 2004, McKay and Fujinaga, 2004, Pampalk et al., 2005]. Others have investigated the ability to search through, annotate, and identify audio files [Dannenberg et al., 2003, Dickerson and Ventura, 2009]. More directly in the realm of computational creativity, researchers have developed systems that can automatically arrange and compose music [de la Puente et al., 2002, Conklin, 2003, Allan and Williams, 2005].

This dissertation presents a computational system that can generate original musical selections that elicit particular emotional or physiological responses. Human subjects were asked to rate a number of musical selections according to emotional content, and biofeedback techniques were employed to measure how these selections affected physiological responses such as breathing, heart rate, and skin temperature. This allowed for the compilation of various corpora of musical selections that tend to elicit particular responses.

These musical corpora were then used as training input for the music composition system. Our system employed n-gram models, Hidden Markov Models, and other statistical distributions based on these corpora to probabilistically generate new selections that could produce similar responses. Neural networks and decision trees were employed to evaluate generated selections based on musicality and effectiveness at eliciting a target response. Empirical studies show that these new selections were generally able to elicit the target emotional or physiological response as effectively as human-composed pieces designed with the same goal in mind.

Generated compositions were analyzed to investigate the contribution of features to the emotional content of music and its ability to affect physiological responses. Efficacy of the system is demonstrated through the generation of soundtracks to accompany stories and music to accompany lyrics.

1.1 Thesis Statement

A compositional system based on statistical models of musical selections and machine learning evaluators will be able to generate original musical compositions that elicit the similar emotions and physiological responses as those evoked by the training selections. This generated music will be as effective as human-composed selection at evoking emotions and in effecting changes in skin resistance, skin temperature, breathing rate, and heart rate.

1.2 Publications

Chapters 2 through 8 of this dissertation consist of five published papers and two that are currently under consideration for publication. Research in the area of automatic music generation is discussed in Chapters 2 through 6. Chapters 7 and 8 describe work on topics related to more general machine learning research.

Chapter 2 presents a system that is able to generate original music capable of eliciting a target emotion. Generated selections are unique, often remarkably musical, and able to elicit particular emotions as effectively as human-composed selections designed to elicit those same emotions.

Chapter 3 expands on the work done in Chapter 2. It frames the knowledge collected by the music-generating system as a cognitive model and identifies features responsible for emotional content in music. It also supplies further evidence of the system's effectiveness, evaluating a larger number of generated selections using surveys that allow for both constrained and free-response answers. Results again demonstrate that the system is capable of generating music that elicits a target emotional response at a level similar to that of human competence at the same task.

Chapter 4 extends the function of the system to generating selections that elicit a particular physiological response. It proves to be as effective as a human composer at generating original compositions that effect changes in skin resistance, skin temperature, breathing rate, and heart rate. The system is particularly adept at composing pieces that elicit target responses in individuals who demonstrated predictable responses to training selections.

More practical effectiveness of the system is demonstrated in subsequent chapters. In Chapter 5, the system is used in conjunction with a textual emotion-labeling system to generate musical accompaniments for fairy tales. Survey data indicates that the addition of music with targeted emotional content makes listening to the stories significantly more enjoyable and increases listener perception of emotion in the text.

Chapter 6 demonstrates how the system can be used to automatically generate and evaluate musical accompaniments for a given set of lyrics. The system proves itself capable of producing melodies in a variety of musical styles. Survey data indicates that generated melodies were often both as pleasant and as good a fit with the text of the lyrics as the original human-composed melodies.

Chapter 7 presents the strategy of Aggregate Certainty Estimators, a technique that uses multiple measures to estimate a classifier's certainty in its prediction on an instance-by-instance basis. These certainty estimators allow the system to outperform a number of baseline strategies including bagging, boosting, stacking, and arbitration in terms of average classification accuracy over 36 data sets.

Chapter 8 provides an analysis of the behavior of the strategy of Bayesian model averaging and explains why it has difficulty achieving high classification accuracy on many empirical tasks. It proposes several different *Bayesian model combination* approaches which allow for more accurate classification behavior. Even the most simplistic of these strategies can compete with traditional *ad hoc* techniques, as well as significantly outperform Bayesian model averaging in most instances.

The following are citations for the papers that comprise the various chapters. References to these papers are numbered here according to the chapter in which they appear.

2. K. Monteith, T. Martinez, and D. Ventura. Automatic Generation of Music for Inducing Emotive Response. In *Proceedings of the First International Conference on Computational Creativity*, pages 140-149, 2010.

3. K. Monteith, T. Martinez, and D. Ventura, Computational Modeling of Emotional Content in Music. In *Proceedings of International Conference on Cognitive Science*, pages 2356-2361, 2010.

4. K. Monteith, B. Brown, D. Ventura, and T. Martinez, Automatic Generation of Music for Inducing Physiological Responses. *In submission.*

5. K. Monteith, V. Francisco, T. Martinez, P. Gervás, and D. Ventura. Automatic Generation of Emotionally-Targeted Soundtracks. In *Proceedings of the Second International Conference on Computational Creativity*, pages 60-62, 2011.

6. K. Monteith, T. Martinez, and D. Ventura. Automatic Generation of Melodic Accompaniments for Lyrics. In *Proceedings of the Third International Conference on Computational Creativity*, pages 87-94, 2012.

7. K. Monteith and T. Martinez. Aggregate Certainty Estimators. To appear in *Computational Intelligence.*

8. K. Monteith, J. Carroll, N. Toronto, K. Seppi, T. Martinez. The Problem with Bayesian Model Averaging (And How to Fix It). *In submission.*

Reports on more preliminary research on the topics covered in Chapters 7 and 8 were published in the following papers:

K. Monteith and T. Martinez. Using Multiple Measures to Predict Confidence in Instance Classification, In *Proceedings of the IEEE International Joint Conference on Neural Networks IJCNN'10*, 4192-4199, 2010.

K. Monteith, J. Carroll, K. Seppi, and T. Martinez. Turning Bayesian Model Averaging into Bayesian Model Combination. In *Proceedings of the IEEE International Joint Conference on Neural Networks IJCNN'11*, 2657-2663, 2011.

Part II

Music-Related Research

”I happen to think that computers are the most important thing to happen to musicians since the invention of cat-gut, which was a long time ago.” –Robert Moog

Chapter 2

Automatic Generation of Music for Inducing Emotive Response

> ***"Music is something that always lifts my spirits and makes me happy, and when I make music I always hope it will have the same effect on whoever listens to it."* –Aaron Carter**

K. Monteith, T. Martinez, and D. Ventura. Automatic Generation of Music for Inducing Emotive Response. In *Proceedings of the First International Conference on Computational Creativity*, pages 140-149, 2010.

Abstract: *We present a system that generates original music designed to match a target emotion. It creates n-gram models, Hidden Markov Models, and other statistical distributions based on musical selections from a corpus representing a given emotion and uses these models to probabilistically generate new musical selections with similar emotional content. This system produces unique and often remarkably musical selections that tend to match a target emotion, performing this task at a level that approaches human competency for the same task.*

2.1 Introduction

Music is a significant creative achievement. Every culture in history has incorporated music into life in some manner. As Wiggins [2006] explains, "musical behavior is a uniquely human trait...further, it is also ubiquitously human: there is no known human society which does not exhibit musical behaviour in some form." Perhaps one of the reasons musical behavior is tied so closely to humanity is its ability to profoundly affect human physiology and emotion. One study found that, when subjects were asked to select music that they found to be particularly pleasurable, listening to this type of music activated the same areas of the brain activated by other euphoric stimuli such as food, sex, or illegal drugs. The authors highlight the significance of the fact that music has an effect on the brain similar to that of "biologically relevant, survival-related stimuli" [Blood and Zatorre, 2001].

Computing that possesses some emotional component, termed affective computing, has received increased attention in recent years. Picard [1995] emphasizes the fact that "emotions play a necessary role not only in human creativity and intelligence, but also in rational human thinking and decision-making. Computers that will interact naturally and intelligently with humans need the ability to at least recognize and express affect." From a theoretical standpoint, it seems reasonable to incorporate emotional awareness into systems designed to mimic (or produce) human-like creativity and intelligence, since emotions are such a basic part of being human. On a more practical level, affective displays on the part of a computerized agent can improve function and usability. Research has shown that incorporating emotional expression into the design of interactive agents can improve user engagement, satisfaction, and task performance [Klein et al., 2002, Partala and Surakka, 2004]. Users may also regard an agent more positively [Ochs et al., 2008] and consider it to be more believable [Bates, 1994] when it demonstrates appropriate emotional awareness.

Given music's ability to alter or heighten emotional states and affect physiological responses, the ability to create music specifically targeted to a particular emotion could have considerable benefits. Calming music can aid individuals in dealing with anxiety disorders or high-anxiety situations. Joyful and energizing music can be a strong motivating force for activities such as exercise and physical therapy. Music therapists use music with varied emotional content in a wide array of musical interventions. The ability to create emotionally-targeted music could also be valuable in creating soundtracks for stories and films.

This paper presents a system that takes emotions into account when creating musical compositions. It produces original music with a desired emotional content using statistical models created from a corpus of songs that evoke the target emotion. Corpora of musical data representing a variety of emotions are collected for use by the system. Melodies are then constructed using n-gram models representing pitch intervals commonly found in the training corpus for a desired emotion. Hidden Markov Models are used to produce harmonies similar to those found in the appropriate corpus. The system also selects the accompaniment pattern and instrumentation for the generated piece based on the likelihood of various accompaniments and instruments appearing in the target corpus. Since it relies entirely on statistics gathered from these training corpora, in one sense the system is learning to imitate emotional musical behavior of other composers when producing its creative works. Survey data indicates that the system composes selections that are as novel and almost as musical as human-composed songs. Without creating any rules for emotional music production, it manages to compose songs that convey a target emotion with surprising accuracy relative to human performance of the same task.

Multiple research agendas bear some relation to our approach. Conklin [2003] summarizes a number of statistical models which can be used for music generation, including random walk, Hidden Markov Models, stochastic sampling, and pattern-based sampling. These approaches can be seen in a number of different studies. For example, Hidden Markov Models have been used to harmonize melodies, considering melodic notes as observed events and a chord progression as a series of hidden states [Allan and Williams, 2005]. Similarly, Markov chains have been used to harmonize given melody lines, focusing on harmonization in a given style in addition to finding highly probable chords [Chuan and Chew, 2007].

Genetic algorithms have also been used in music composition tasks. De la Puente and associates [2002] use genetic algorithms to learn melodies, employing a fitness function that considers differences in pitch and duration in consecutive notes. Horner and Goldberg [1991] attempt to create more cohesive musical selections using a fitness function that evaluates generated phrases according agreement with a thematic phrase. Tokui and Iba [2000] focus their attention on using genetic algorithms to learn polyphonic rhythmic patterns, evaluating patterns with a neural network that learns to predict which patterns the user would most likely rate highly.

Musical selections can also be generated through a series of musical grammar rules. These rules can either be specified by an expert or determined by statistical models. For example, Ponsford, Wiggins, and Mellish [1998] use n-gram statistical methods for learning musical grammars. Phon-Amnuaisuk and Wiggins [1999] compare genetic algorithms to a rule-based approach for the task of four-part harmonization.

Delgado, Fajardo, and Molina-Solana [2009] use a rule-based system to generate compositions according to a specified mood. Rutherford and Wiggins analyze the features that contribute to the emotion of fear in a musical selection and present a system that allows for an input parameter that determines the level of "scariness" in the piece [Rutherford and Wiggins, 2003]. Oliveira and Cardoso [2007] describe a wide array of features that contribute to emotional content in music and present a system that uses this information to select and transform chunks of music in accordance with a target emotion.

Like these previously mentioned systems, our system is concerned with producing music with a desired emotional content. It employs a number of statistical methods discussed in the previously mentioned papers. Rather than developing rule sets for different emotions, it composes original music based on statistical information in training corpora.

2.2 Methodology

In order to produce selections with specific emotional content, a separate set of musical selections is compiled for each desired emotion. Initial experiments focus on the six basic emotions outlined by Parrott [2001]—love, joy, surprise, anger, sadness, and fear—creating a data set representative of each. Selections for the training corpora are taken from movie soundtracks due to the wide emotional range present in this genre of music. MIDIs used in the experiments can be found at the Free MIDI File Database.[1] These MIDIs were rated by

[1] http://themes.mididb.com/movies/

a group of research subjects. Each selection was rated by at least six subjects, and selections rated by over 80% of subjects as representative of a given emotion were then selected for use in the training corpora.

Next, the system analyzes the selections to create statistical models of the data in the six corpora. Selections are first transposed into the same key. Melodies are then analyzed and n-gram models are generated representing what notes are most likely to follow a given series of notes in a given corpus. Statistics describing the probability of a melody note given a chord, and the probability of a chord given the previous chord, are collected for each of the six corpora. Information is also gathered about the rhythms, the accompaniment patterns, and the instrumentation present in the songs.

Since not every melody produced is likely to be particularly remarkable, the system also makes use of multilayer perceptrons with a single hidden layer to evaluate the generated selections. Inputs to these neural networks are the default features extracted by the "Phrase Analysis" component of the freely available jMusic software.[2] This component returns a vector of twenty-one statistics describing a given melody, including factors such as number of consecutive identical pitches, number of distinct rhythmic values, tonal deviation, and key-centeredness.

A separate set of two networks is developed to evaluate both generated rhythms and generated pitches. The first network in each set is trained using analyzed selections in the target corpus as positive training instances and analyzed selections from the other corpora as negative instances. This is intended to help the system distinguish selections containing the desired emotion. The second network in each set is trained with melodies from all corpora versus melodies previously generated by the algorithm. In this way, the system learns to emulate melodies which have already been accepted by human audiences.

Once the training corpora are set and analyzed, the system employs four different components: a Rhythm Generator, a Pitch Generator, a Chord Generator, and an Accom-

[2] http://jmusic.ci.qut.edu.au/

paniment and Instrumentation Planner. The functions of these components are explained in more detail in the following sections.

2.2.1 Rhythm Generator

The rhythm for the selection with a desired emotional content is generated by selecting a phrase from a randomly chosen selection in the corresponding data set. The rhythmic phrase is then altered by selecting and modifying a random number of measures. The musical forms of all the selections in the corpus are analyzed, and a form for the new selection is drawn from a distribution representing these forms. For example, a very simple AAAA form, where each of four successive phrases contains notes with the same rhythm values, tends to be very common. Each new rhythmic phrase is analyzed by jMusic and then provided as input to the neural network rhythm evaluators. Generated phrases are only accepted if they are classified positively by both neural networks.

2.2.2 Pitch Generator

Once the rhythm is determined, pitches are selected for the melodic line. These pitches are drawn according to the n-gram model constructed from the melody lines of the corpus with the desired emotion. A melody is initialized with a series of random notes, selected from a distribution that model which notes are most likely to begin musical selections in the given corpus. Additional notes in the melodic sequence are randomly selected based on a probability distribution of what note is most likely to follow the given series of n notes. The system generates several hundred possible series of pitches for each rhythmic phrase. As with the rhythmic component, features are then extracted from these melodies using jMusic and provided as inputs to the neural network pitch evaluators. Generated melodies are only selected if they are classified positively by both neural networks.

2.2.3 Chord Generator

The underlying harmony is determined using a Hidden Markov Model, with pitches considered as observed events and the chord progression as the underlying state sequence. The Hidden Markov Model requires two conditional probability distributions: the probability of a melody note given a chord and the probability of a chord given the previous chord. The statistics for these probability distributions are gathered from the corpus of music representing the desired emotion. The system then calculates which set of chords is most likely given the melody notes and the two conditional probability distributions. Since many of the songs in the training corpora had only one chord present per measure, initial attempts at harmonization also make this assumption, considering only downbeats as observed events in the model.

2.2.4 Accompaniment and Instrumentation Planner

The accompaniment patterns for each of the selections in the various corpora are categorized, and the accompaniment pattern for a generated selection is probabilistically selected from the patterns of the target corpus. Common accompaniment patterns included arpeggios, chords sounding on repeated rhythmic patterns, and a low base note followed by chords on non-downbeats. (A few of the accompaniment patterns such as *Star Wars: Duel of the Fates* and *Addams Family* had to be rejected or simplified; they were so characteristic of the training selections that they were too recognizable in the generated song.) Instruments for the melody and harmonic accompaniment are also probabilistically selected based on the frequency of various melody and harmony instruments in the corpus.

2.3 Results

Colton [2008] suggests that, for a computational system to be considered creative, it must be perceived as possessing skill, appreciation, and imagination. The system could be considered "skillful" if it demonstrates knowledge of traditional music behavior. This is accomplished by taking advantage of statistical knowledge to train the system to behave according to traditional musical conventions. The system may be considered "appreciative" if it can produce something of value and adjust its work according to the preferences of itself or others. This is addressed through the neural networks evaluators. The "imaginative" criterion can be met if the system can create new material independent of both its creators and other composers. Since all of the generated songs can be distinguished from the songs in the training corpora, this criterion is met at least on a basic level. However, to further evaluate all of these aspects, the generated songs were subjected to human evaluation. Twelve selections were generated for testing purposes.[3] Each selection was then played for thirteen individuals, who were asked to answer the following questions:

1. What emotions are present in this selection (circle all that apply)?
2. On a scale of one to ten, how much does this sound like real music?
3. On a scale of one to ten, how unique does this selection sound?

The first two questions target the aspects of skill and appreciation, ascertaining whether the system is skillful enough to produce something both musical and representative of a given emotion. The third question evaluates the imagination of the system, determining whether or not the generated music is perceived as novel by human audiences.

To provide a baseline, two members of the campus songwriting club were asked to perform the same task as the computer: compose a musical selection representative of one of six given emotions. Each composer provided three songs. These selections were also played and subjects were asked to evaluate them according to the same three questions. Song order was randomized, and while subjects were told that some selections were written by a computer and some by a human, they were not told which selections belonged to which categories.

[3] These selections are available at http://axon.cs.byu.edu/emotiveMusicGeneration

Table 2.1: Emotional Content of Computer-Generated Music. Percentage of survey respondents who identified a given emotion for songs generated in each of the six categories. The first column provides the categories of emotions for which songs were generated. Column headers describe the emotions identified by survey respondents.

	Love	Joy	Surprise	Anger	Sadness	Fear
Love	**0.62**	0.92	0.08	0.00	0.00	0.00
Joy	0.38	**0.69**	0.15	0.00	0.08	0.08
Surprise	0.08	0.46	**0.62**	0.00	0.00	0.00
Anger	0.00	0.00	0.08	**0.46**	0.38	0.69
Sadness	0.09	0.18	0.27	0.18	**0.45**	0.36
Fear	0.15	0.08	0.00	0.23	0.62	**0.23**

Table 2.2: Emotional Content of Human-Generated Music. Percentage of survey respondents who identified a given emotion for songs composed in each of the six categories.

	Love	Joy	Surprise	Anger	Sadness	Fear
Love	**0.64**	0.64	0.00	0.09	0.09	0.00
Joy	0.77	**0.31**	0.15	0.00	0.31	0.00
Surprise	0.00	0.27	**0.18**	0.09	0.45	0.27
Anger	0.00	0.09	0.18	**0.27**	0.73	0.64
Sadness	0.38	0.08	0.00	0.00	**0.77**	0.08
Fear	0.09	0.00	0.00	0.27	0.55	**0.45**

Table 2.1 reports on how survey participants responded to the first question. It gives the percentage of respondents who identified a given emotion in computer-generated selections in each of the six categories. Table 2.2 provides a baseline for comparison by reporting the same data for the human-generated pieces. Tables 2.3 and 2.4 address the second two survey questions. They provide the average score for musicality and novelty (on a scale from one to ten) received by the various selections.

In all cases, the target emotion ranked highest or second highest in terms of the percentage of survey respondents identifying that emotion as present in the computer-generated songs. In four cases, it was ranked highest. Respondents tended to think that the love songs sounded a little more like joy than love, and that the songs portraying fear sounded a little sadder than fearful. But surprisingly, the computer-generated songs appear to be slightly better at communicating an intended emotion than the human-generated songs. Averaging

Table 2.3: Musicality and Novelty of Computer-Generated Music. Average score (on a scale of one to ten) received by selections in the various categories in response to survey questions about musicality and novelty.

	Musicality	Novelty
Love	8.35	4.12
Joy	6.28	5.86
Surprise	6.47	4.78
Anger	5.64	4.96
Sadness	7.09	4.40
Fear	6.53	5.07
Average:	6.73	4.86

Table 2.4: Musicality and Novelty of Human-Generated Music. Average score (on a scale of one to ten) received by selections in the various categories in response to survey questions about musicality and novelty.

	Musicality	Novelty
Love	7.73	4.45
Joy	9.15	4.08
Surprise	7.09	5.36
Anger	8.18	4.60
Sadness	9.23	4.08
Fear	5.45	5.45
Average:	7.81	4.67

over all categories, 54% of respondents correctly identified the target emotion in computer-generated songs, while only 43% of respondents did so for human-generated songs.

Human-generated selections did tend to sound more musical, averaging a 7.81 score for musicality on a scale of one to ten as opposed to the 6.73 scored by computer-generated songs. However, the fact that a number of the computer-generated songs were rated as more musical than the human-produced songs is somewhat impressive. Computer-generated songs were also rated on roughly the same novelty level as the human-generated songs, receiving a 4.86 score as opposed to the human score of 4.67. As an additional consideration, the computer-generated songs were produced in a more efficient and timely manner than the human-generated ones. Only one piece in each category was submitted for survey purposes due to the difficulty of finding human composers with the time to provide music for this project.

2.4 Discussion and Future Work

Pearce, Meredith, and Wiggins [2002] suggest that music generation systems concerned with the computational modeling of music cognition be evaluated both by the music they produce and by their behavior during the composition process. The system discussed here can be considered creative both in the fact that it can produce fairly high-quality music, and that it does so in a creative manner. In *Creativity: Flow and the Psychology of Discovery and Invention* (Chapter 2), Csikszentmihalyi [1996] includes several quotes by the inventor Rabinow outlining three components necessary for being a creative, original thinker. The system described in this work meets all three criteria for creativity.

As Rabinow explains, "First, you have to have a tremendous amount of information...If you're a musician, you should know a lot about music..." Computers have a unique ability to store and process large quantities of data. They have the potential even to have some advantage over humans in this particular aspect of the creative process if the knowledge can be collected, stored, and utilized effectively. The system discussed in this paper addresses this aspect of the creative process by gathering statistics from the various corpora of musical selections and using this information to inform choices about rhythm, pitch, and harmony.

The next step is generation based on the domain information. Rabinow continues: "Then you have to be willing to pull the ideas...come up with something strange and different." The system described in this work can create a practically unlimited number of unique melodies based on random selections from probability distributions. Again, computers have some advantage in this area. They can generate original music quickly and tirelessly. Some humans have been able to produce astonishing numbers of compositions; Bach's work alone fills sixty volumes. But while computers are not yet producing original work of Bach's creativity and caliber, they could easily outdistance him in sheer output.

The final step is evaluation of these generated melodies, Rabinow's third suggestion: "And then you must have the ability to get rid of the trash which you think of. You cannot think only of good ideas, or write only beautiful music..." Our system addresses this aspect through the neural network evaluators. It learns to select pieces with features similar to musical selections that have already been accepted by human audiences and ones most like selections humans have labeled as expressing a desired emotion. It even has the potential to improve over time by producing more negative examples and learning to distinguish these from positive ones. But finding good features for use in the evaluating classifiers poses a significant challenge. First attempts at improving the system will involve modifications in this area.

As previously mentioned, research has been done to isolate specific features that are likely responsible for the emotional content of a song [Rutherford and Wiggins, 2003, Oliveira and Cardoso, 2007]. Incorporating such features into the neural network evaluators could provide these evaluators with significantly more power in selecting the melodies most representative of a desired emotion. Despite the possible improvements, it is quite encouraging to note that even naïve evaluation functions are able to produce fairly musical and emotionally targeted selections.

Additional improvements will involve drawing from a larger corpus of data for song generation. Currently, the base seems to be sufficiently wide to produce songs that were considered to be as original as human-composed songs. However, many of the generated pieces tend to sound somewhat similar to each other. On the other hand, sparseness of training data actually provides some advantages. For example, in some cases, the presence of fewer examples in the training corpus resulted in similar musical motifs in the generated songs. Phrases would often begin with the same few notes before diverging, particularly in corpora where songs tended to start on the same pitch of the scale. Larger corpora will allow for the generation of more varied songs, but to maintain musicality, the evaluation mechanism might be extended to encourage the development of melodic motifs among the various phrases.

The type and magnitude of emotions can often be indicated by concurrent physiological responses. The format of these experiments lends itself to the additional goal of generating music targeted to elicit a desired physiological response. Future work will involve measuring responses such as heart rate, muscle tension, and skin conductance and how these are affected by different musical selections. This information could then be used to create training corpora of songs likely to produce desired physiological responses. These could then be used to generate songs with similar properties. The format also allows for the generation of songs that can switch emotions at a desired point in time simply by switching to using statistical data from a different corpus.

The system described here is arguably creative by reasonable standards. It follows a creative process as suggested by Rabinow and others, producing and evaluating reasonably skillful, novel, and emotionally targeted compositions. However, our system will really only be useful to society if it produces music that not only affects emotions, but that people will listen to long enough for that effect to take place. This is difficult to demonstrate in a short-term evaluation study, but we do appear to be on the right track. A few of the generated pieces received musicality ratings similar to those of the human-produced pieces. Many of those surveyed were surprised that the selections were written by a computer. Another survey respondent announced that the program had “succeeded” because one of the computer-generated melodies had gotten stuck in his head. These results show promise for the possibility of producing a system that is truly creative.

Acknowledgments

This work is supported by the National Science Foundation under Grant No. IIS-0856089. Special thanks to Heather Hogue and Paul McFate for providing the human-generated music.

Chapter 3

Computational Modeling of Emotional Content in Music

"Computers are famous for being able to do complicated things starting from simple programs." **–Seth Lloyd**

K. Monteith, T. Martinez, and D. Ventura, Computational Modeling of Emotional Content in Music. In *Proceedings of International Conference on Cognitive Science*, pages 2356-2361, 2010.

Abstract: *We present a system designed to model characteristics which contribute to the emotional content of music. It creates n-gram models, Hidden Markov Models, and entropy-based models from corpora of musical selections representing various emotions. These models can be used both to identify emotional content and generate pieces representative of a target emotion. According to survey results, generated selections were able to communicate a desired emotion as effectively as human-generated compositions.*

3.1 Introduction

Music and emotion are intrinsically linked; music is able to express emotions that cannot adequately be expressed by words alone. Often, there is strong consensus among listeners as to what type of emotion is being expressed in a particular piece [Gabrielsson and Lindstrom, 2001, Juslin, 2001]. There is even some evidence to suggest that some perceptions of emotion in music may be innate. For example, selections sharing some acoustical properties of fear vocalizations, such as sudden onset, high pitch, and strong energy in the high frequency range, often provoke physiological defense responses [Ohman, 1988]. Researchers have demonstrated similar low-level detection mechanisms for both pleasantness and novelty. [Scherer, 1984, 1988]. There also appears to be some inborn preference for consonance over dissonance. In studies with infants, researchers found that their subjects looked significantly longer at the source of sound and were less likely to squirm and fret when presented with consonant as opposed to dissonant versions of a melody [Zentner and Kagan, 1996].

There are a variety of theories as to what aspects of music are most responsible for eliciting emotional responses. Meyer [1956] theorizes that meaning in music comes from following or deviating from an expected structure. Sloboda [1985] emphasizes the importance of associations in the perception of emotion in music and gives particular emphasis to association with lyrics as a source for emotional meaning. Kivy [1980] argues for the importance of cultural factors in understanding emotion and music, proposing that the "emotive life" of a culture plays a major role in the emotions that members of that culture will detect in their music. Tolbert [2001] proposes that children learn to associate emotion with music in much the same way that they learn to associate emotions with various facial expressions. Scherer [2001] presents a framework for formally describing the emotional effects of music and then outlines factors that contribute to these emotions, including structural, performance, listener, and contextual features.

In this paper, we focus on some of the structural aspects of music and the manner in which they contribute to emotions in music. We present a cognitive model of characteristics of music responsible for human perception of emotional content. Our model is both discriminative and generative; it is capable of detecting a variety of emotions in musical selections, and also of producing music targeted to a specific emotion.

3.2 Related Work

A number of researchers have addressed the task of modeling musical structure for the purposes of building a generative musical system. Conklin [2003] summarizes a number of statistical models which can be used for music generation, including random walk, Hidden Markov Models, stochastic sampling, and pattern-based sampling. These approaches can be seen in a number of different studies. For example, Hidden Markov Models have been used to harmonize melodies, considering melodic notes as observed events and a chord progression as a series of hidden states [Allan and Williams, 2005]. Similarly, Markov chains have been used to harmonize given melody lines, focusing on harmonization in a given style in addition to finding highly probable chords [Chuan and Chew, 2007].

Wiggins, Pearce, and Mullensiefen [2009] present a system designed to model factors such as pitch expectancy and melodic segmentation. They also demonstrate that their system can successfully generate music in a given style. Systems have also been developed to produce compositions with targeted emotional content. Delgado, Fajardo, and Molina-Solana [2009] use a rule-based system to generate compositions according to a specified mood. Rutherford and Wiggins [2003] analyze the features that contribute to the emotion of fear in a musical selection and present a system that allows for an input parameter that determines the level of "scariness" in the piece. Oliveira and Cardoso [2007] describe a wide array of features that contribute to emotional content in music and present a system that uses this information to select and transform chunks of music in accordance with a target emotion. The authors have also developed a system that addresses the task of composing music with a specified emotional content [Monteith et al., 2010]. In this paper, we illustrate how our system can be interpreted as a cognitive model of human perception of emotional content in music.

3.3 Methodology

The proposed system constructs statistical and entropic models for various emotions based on corpora of human-labeled musical data. Analysis of these models provides insights as to why certain music evokes certain emotions. The models supply localized information about intervals and chords that are more common to music conveying a specific emotion. They also supply information about what overall melodic characteristics contribute to emotional content. To validate our findings, we generate a number of musical selections and ask research subjects to label the emotional content of the generated music. Similar experiments are conducted with human-generated music commissioned for the project. We then observe the correlations between subject responses and our predictions of emotional content.

Initial experiments focus on the six basic emotions outlined by Parrott [2001]—love, joy, surprise, anger, sadness, and fear—creating a data set representative of each. A separate set of musical selections is compiled for each of the emotions studied. Selections for the training corpora are taken from movie soundtracks due to the wide emotional range present in this genre of music. MIDI files used in the experiments can be found at the Free MIDI File Database.[1] These MIDI files were rated by a group of research subjects. Each selection was rated by at least six subjects, and selections rated by over 80% of subjects as representative of a given emotion were then selected for use in the training corpora. Selections used for these experiments are shown in Figure 3.1.

Next, the system analyzes the selections to create statistical models of the data in the six corpora. Selections are first transposed into the same key. Melodies are then analyzed and n-gram models are generated representing what notes are most likely to follow a given series of notes in a given corpus. Statistics describing the probability of a melody note given a chord, and the probability of a chord given the previous chord, are collected for each of the six corpora. Information is also gathered about the rhythms, the accompaniment patterns, and the instrumentation present in the songs.

The system also makes use of decision trees constructed to model the characteristics that contribute to emotional content. These trees are constructed using the C4.5 algorithm [Quinlan, 1993], an extension of the ID3 algorithm [Quinlan, 1986] that allows for real-valued attributes. The decision tree classifiers allow for a more global analysis of generated melodies. Inputs to these classifiers are the default features extracted by the "Phrase Analysis" component of the freely available jMusic software.[2] This component returns a vector of twenty-one

[1] http://themes.mididb.com/movies/

[2] http://jmusic.ci.qut.edu.au/

Love:	**Joy:**
Advance to the Rear	1941
Bridges of Madison County	633 Squadron
Casablanca	Baby Elephant Walk
Dr. Zhivago	Chariots of Fire
Legends of the Fall	Flashdance
Out of Africa	Footloose
Surprise:	Jurassic Park
Addams Family	Mrs. Robinson
Austin Powers	That Thing You Do
Batman	You're the One that I Want
Dueling Banjos	**Anger:**
George of the Jungle	Gonna Fly Now
Nightmare Before Christmas	James Bond
Pink Panther	Mission Impossible
The Entertainer	Phantom of the Opera
Toy Story	Shaft
Willie Wonka	**Fear:**
Sadness:	Axel's Theme
Forrest Gump	Beetlejuice
Good Bad Ugly	Edward Scissorhands
Rainman	Jaws
Romeo and Juliet	Mission Impossible
Schindler's List	Phantom of the Opera
	Psycho
	Star Wars: Duel of the Fates
	X-Files: The Movie

Figure 3.1: Selections used in training corpora for the six different emotions considered.

statistics describing a given melody, including factors such as number of consecutive identical pitches, number of distinct rhythmic values, tonal deviation, and key-centeredness. These statistics are calculated for both the major and minor scales.

A separate set of classifiers is developed to evaluate both generated rhythms and generated pitches. The first classifier in each set is trained using analyzed selections in the target corpus as positive training instances and analyzed selections from the other corpora as negative instances. This is intended to help the system distinguish selections containing the desired emotion. The second classifier in each set is trained with melodies from all corpora versus melodies previously generated by the algorithm, allowing the system to learn melodic characteristics of selections which have already been accepted by human audiences.

For the generative portion of the model, the system employs four different components: a Rhythm Generator, a Pitch Generator, a Chord Generator, and an Accompaniment and Instrumentation Planner. The functions of these components are explained in more detail in the following sections.

3.3.1 Rhythm Generator

The rhythm for the selection with a desired emotional content is generated by selecting a phrase from a randomly chosen selection in the corresponding data set. The rhythmic phrase is then altered by selecting and modifying a random number of measures. The musical forms of all the selections in the corpus are analyzed, and a form for the new selection is drawn from a distribution representing these forms. For example, a very simple AAAA form, where each of four successive phrases contains notes with the same rhythm values, tends to be very common. Each new rhythmic phrase is analyzed by jMusic and then provided as input to the rhythm evaluators. Generated phrases are only accepted if they are classified positively by both classifiers.

3.3.2 Pitch Generator

Once the rhythm is determined, pitches are selected for the melodic line. These pitches are drawn according to the n-gram model constructed from melody lines of the corpus with the desired emotion. A melody is initialized with a series of random notes, selected from a distribution that models notes most likely to begin musical selections in the given corpus. Additional notes in the melodic sequence are randomly selected based on a probability distribution of note most likely to follow the given series of n notes.

For example, with the “joy” corpus, the note sequence (C4, D4, E4) has a 0.667 probability of being followed by an F4, a 0.167 probability of being followed by a D4, and a

0.167 probability of being followed by a C4. If these three notes were to appear in succession in a generated selection, the system would have a 0.167 probability of selecting a C4 as the next note.

The system generates several hundred possible series of pitches for each rhythmic phrase. As with the rhythmic component, features are then extracted from these melodies using jMusic and provided as inputs to the pitch evaluators. Generated melodies are only selected if they are classified positively by both classifiers.

3.3.3 Chord Generator

The underlying harmony is determined using a Hidden Markov Model, with pitches considered as observed events and the chord progression as the underlying state sequence [Rabiner, 1989]. The Hidden Markov Model requires two conditional probability distributions: the probability of a melody note given a chord and the probability of a chord given the previous chord. The statistics for these probability distributions are gathered from the corpus of music representing the desired emotion.

For example, C4 is most likely to be accompanied by a C major chord, and F4 is most likely to be accompanied by a G7 chord in selections from the "love" corpus (probabilities of 0.099 and 0.061, respectively). In the "sadness" corpus, C4 is most likely to be accompanied by a C minor chord (probability of 0.060). As examples from the second set of distributions, the G7 chord is most likely to be followed by the G7 or the C major chord in selections from the "love" corpus (both have a probability of 0.105). In selections from the "sadness" corpus, the G7 chord is most likely to be followed by the G7 or the C minor chord (probabilities of 0.274 and 0.094 respectively).

The system then calculates which set of chords is most likely given the melody notes and the two conditional probability distributions. Since many of the songs in the training corpora had only one chord present per measure, initial attempts at harmonization also make this assumption, considering only downbeats as observed events in the model.

3.3.4 Accompaniment and Instrumentation Planner

The accompaniment patterns for each of the selections in the various corpora are categorized, and the accompaniment pattern for a generated selection is probabilistically selected from the patterns of the target corpus. Common accompaniment patterns included arpeggios, block chords sounding on repeated rhythmic patterns, and a low base note followed by chords on non-downbeats.

For example, arpeggios are a common accompaniment pattern in the corpus of selections expressing the emotion of “love.” Two of the selections in the corpus feature simple, arpeggiated chords as the predominant theme in their accompaniments, and two more selections have an accompaniment pattern that feature arpeggiated chords played by one instrument and block chords played by a different instrument. The remaining two selections in the corpus feature an accompaniment pattern of a low base note followed by chords on non-downbeats. When a new selection is generated by the system, one of these three patterns is selected with equal likelihood to be the accompaniment for the new selection.

Instruments for the melody and harmonic accompaniment are also probabilistically selected based on the frequency of various melody and harmony instruments in the corpus. For example, melody instruments for selections in the “surprise” corpus include acoustic grand piano, electric piano, and piccolo. Harmony instruments include trumpet, trombone, acoustic grand piano, and acoustic bass.

3.3.5 Evaluation

In order to verify that our system was accurately modeling characteristics contributing to emotional content, we presented our generated selections to research subjects and asked them to identify the emotions present. Forty-eight subjects, ages 18 to 55, participated in this study. Six selections were generated in each category, and each selection was played for four subjects. Subjects were given the list of emotions and asked to circle all emotions that were represented in each song. Each selection was also played for four subjects who had not seen the list of emotions. These subjects were asked to write down any emotions they thought were present in the music without any suggestions of emotional categories on the part of the researchers. Reported results represent percentages of the twenty-four responses in each category. To provide a baseline, two members of the campus songwriting club were also asked to perform the same task: compose a musical selection representative of one of six given emotions. Each composer provided selections for three of the emotional categories. These selections were evaluated in the same manner as the computer-generated selections, with four subjects listening to each selection for each type of requested response. Reported results represent percentages of the four responses in each category.

3.4 Results

Figures 3.2 through 3.7 outline the characteristics identified by the decision trees as being responsible for emotional content. For example, if a piece had a Dissonance measure over 0.107 and a Repeated Pitch Density measure over 0.188, it was classified in the "anger" category. Informally, angry selections tend to be dissonant and have many repeated notes. Similar information was collected for each of the different emotions. Selections expressing "love" tend to have lower repeated pitch density and fewer repeated patterns of three, indicating these selections tend to be more "flowing." Joyful selections have some stepwise movement in a major scale and tend to have a strong climax at the end. The category of "surprise" appears to be the least cohesive; it requires the most complex set of rules for determining membership in the category. However, repeated pitch patterns of four are present in all the surprising selections, as is a lack of stepwise movement in the major scale. Not surprisingly, selections expressing "sadness" adhere to a minor scale and tend to have a downward trend in pitch. Fearful selections deviate from the major scale, do not always compensate for leaps, and have an upward pitch direction. Downward melodic trends do not deviate as much from the major scale. Our model appears to be learning to detect the melodic minor scale;

```
RepeatedPitchDensity <= 0.146
- RepeatedPitchPatternsOfThree <= 0.433: Yes
- RepeatedPitchPatternsOfThree > 0.433: No
RepeatedPitchDensity > 0.146: No
```

Figure 3.2: Decision tree models of musical characteristics contributing to the emotion of love.

```
PitchMovementByTonalStep <= 0.287: No
PitchMovementByTonalStep > 0.287
- ClimaxPosition <= 0.968
- - ClimaxTonality <= 0: No
- - ClimaxTonality > 0
- - - PitchMovementByTonalStep(Minor) <= 0.535: No
- - - PitchMovementByTonalStep(Minor) > 0.535: Yes
- ClimaxPosition > 0.968: Yes
```

Figure 3.3: Decision tree models of musical characteristics contributing to the emotion of joy.

melodies moving downward in this scale will have a raised sixth and seventh tone, so they differ in only one tone from a major scale.

Tables 3.1 and 3.2 report results for the constrained response surveys. Row labels indicate the corpus used to generate a given selection, and column labels indicate the emotion identified by survey respondents. Based on the results in Table 3.1, our system is successful at modeling and generating music with targeted emotional content. For all of the emotional categories but “surprise,” a majority of people identified the emotion when presented with a list of six emotions. In all cases, the target emotion ranked highest or second highest in terms of the percentage of survey respondents identifying that emotion as present in the computer-generated songs. As a general rule, people were more likely to select the categories of “joy” or “sadness” than some of the other emotions, perhaps because music in western culture is traditionally divided up into categories of major and minor. A higher percentage

```
RepeatedPitchPatternsOfFour <= 0.376: No
RepeatedPitchPatternsOfFour > 0.376
- PitchMovementByTonalStep (Minor) <= 0.550
- - ClimaxPosition <= 0.836: Yes
- - ClimaxPosition > 0.836
- - - LeapCompensation <= 0.704: No
- - - LeapCompensation > 0.704
- - - - KeyCenteredness <= 0.366: No
- - - - KeyCenteredness > 0.366: Yes
- PitchMovementByTonalStep(Minor) > 0.550: No
```

Figure 3.4: Decision tree models of musical characteristics contributing to the emotion of surprise.

```
Dissonance <= 0.107: No
Dissonance > 0.107
- RepeatedPitchDensity <= 0.188: No
- RepeatedPitchDensity > 0.188: Yes
```

Figure 3.5: Decision tree models of musical characteristics contributing to the emotion of anger.

of people identified “joy” in songs designed to express “love” or “surprise” than identified the target emotion. “Fear” was also a commonly selected category. More people identified angry songs as fearful, perhaps due to the sheer amount of scary-movie soundtracks in existence. Themes from “Jaws,” “Twilight Zone,” or “Beethoven’s Fifth Symphony” readily come to mind as appropriate music to accompany frightening situations; thinking of an iconic song in the “anger” category is more of a challenging task. Averaging over all categories, 57.67% of respondents correctly identified the target emotion in computer-generated songs, while only 33.33% of respondents did so for the human-generated songs.

For the open-ended questions, responses were evaluated by similarity to Parrott’s expanded hierarchy of emotions. Each of the six emotions can be broken down into a number of secondary emotions, which can in turn be subdivided into tertiary emotions. If a word

```
TonalDeviation(Minor) <= 0.100
- OverallPitchDirection <= 0.500: Yes
- OverallPitchDirection > 0.500: No
TonalDeviation (Minor) > 0.100: No
```

Figure 3.6: Decision tree models of musical characteristics contributing to the emotion of sadness.

```
TonalDeviation <= 0.232: No
TonalDeviation > 0.232
- LeapCompensation <= 0.835
- - OverallPitchDirection <= 0.506
- - - TonalDeviation <= 0.290: Yes
- - - TonalDeviation > 0.290: No
- - OverallPitchDirection > Yes
- LeapCompensation > 0.835: No
```

Figure 3.7: Decision tree models of musical characteristics contributing to the emotion of fear.

in the subject's response matched any form of one of these primary, secondary, or tertiary emotions, it was categorized as the primary emotion of the set. Results are reported in Tables 3.3 and 3.4. Again, row labels indicate the corpus used to generate a given selection, and column labels indicate the emotion identified by survey respondents.

The target emotion also ranked highest or second highest in terms of the percentage of survey respondents identifying that emotion as present in the computer-generated songs for the open-ended response surveys. Without being prompted or limited to specific categories, and with a rather conservative method of classifying subject response, listeners were still often able to detect the original intended emotion. Once again, the computer-generated songs appear to be slightly more emotionally communicative. 21.67% of respondents correctly identified the target emotion in computer-generated songs in these open-ended surveys, while only 16.67% of respondents did so for human-generated songs.

Table 3.1: Emotional Content of Computer-Generated Music. Percentage of survey respondents who identified a given emotion for selections generated in each of the six categories. Row labels indicate the corpus used to generate a given selection, and column labels indicate the emotion identified by survey respondents.

	Love	Joy	Surprise	Anger	Sadness	Fear
Love	58%	75%	12%	4%	21%	0%
Joy	58%	88%	25%	0%	4%	0%
Surprise	4%	54%	38%	0%	12%	8%
Anger	4%	04%	46%	50%	17%	88%
Sadness	0%	8%	25%	42%	62%	58%
Fear	17%	21%	29%	12%	67%	50%

Table 3.2: Emotional Content of Human-Generated Music.

	Love	Joy	Surprise	Anger	Sadness	Fear
Love	50%	0%	25%	25%	100%	0%
Joy	100%	25%	0%	0%	75%	0%
Surprise	0%	0%	50%	75%	50%	50%
Anger	25%	25%	0%	25%	50%	50%
Sadness	75%	25%	25%	25%	0%	25%
Fear	50%	0%	0%	0%	100%	50%

Listeners cited “fondness,” “amorousness,” and in one rather specific case, “unrequited love,” as emotions present in selections from the “love” category. One listener said it sounded like “I just beat the game.” Another mentioned “talking to Grandpa” as a situation the selection called to mind. Reported descriptions of selections in the “joy” category most closely matched Parrott’s terms. These included words such as “happiness,” “triumph,” “excitement”, and “joviality.” Selections were also described as “adventurous” and “playful.”

None of the songs in the category of “surprise” were described using Parrott’s terms. However, this is not entirely unexpected considering the fact that Parrott lists a single secondary emotion and three tertiary emotions for this category. By comparison, the category of joy has six secondary emotions and 34 tertiary emotions. The general sentiment of “surprise” still appears to be present in the responses. One listener reported that the selection sounded

Table 3.3: Emotional Content of Computer-Generated Music: Unconstrained Responses.

	Love	Joy	Surprise	Anger	Sadness	Fear
Love	21%	25%	0%	0%	0%	0%
Joy	0%	58%	0%	4%	0%	0%
Surprise	0%	12%	0%	8%	0%	0%
Anger	0%	8%	0%	17%	0%	25%
Sadness	4%	0%	0%	4%	17%	17%
Fear	0%	8%	0%	12%	17%	17%

Table 3.4: Emotional Content of Human-Generated Music: Unconstrained Responses.

	Love	Joy	Surprise	Anger	Sadness	Fear
Love	0%	25%	0%	0%	0%	0%
Joy	0%	25%	0%	0%	0%	0%
Surprise	0%	0%	0%	0%	25%	0%
Anger	0%	0%	0%	0%	25%	0%
Sadness	0%	0%	0%	0%	25%	0%
Fear	0%	0%	0%	25%	25%	50%

like an ice cream truck. Another said it sounded like being literally drunken with happiness. "Playfulness," "childishness," and "curiosity" were also used to describe the selections.

Angry songs were often described using Parrott's terms of "annoyance" and "agitation." Other words used to describe angry songs included "uneasy," "insistent," and "grim." Descriptions for songs in the "sad" category ranged from "pensive" and "antsy" to "deep abiding sorrow." A few listeners described a possible situation instead of an emotion: "being somewhere I should not be" or "watching a dog get hit by a car." Fearful songs were described with words such as "tension," "angst," and "foreboding." "Hopelessness" and even "homesickness" were also mentioned.

3.5 Conclusion

Pearce, Meredith, and Wiggins [2002] suggest that music generation systems concerned with the computational modeling of music cognition be evaluated both by their behavior during the composition process and by the music they produce. Our system is able to successfully develop cognitive models and use these models to effectively generate music. Just as humans listen to and study the works of previous composers before creating their own compositions, our system learns from its exposure to emotion-labeled musical data. Without being given a set of preprogrammed rules, the system is able to develop internal models of musical structure and characteristics that contribute to emotional content. These models are used both to generate musical selections and to evaluate them before they are output to the listener. The quality of these models is evidenced by the system's ability to produce songs with recognizable emotional content. Results from both constrained and unconstrained surveys demonstrate that the system can accomplish this task as effectively as human composers.

Chapter 4

Automatic Generation of Music for Inducing Physiological Responses

"I think music in itself is healing. It's an explosive expression of humanity. It's something we are all touched by. No matter what culture we're from, everyone loves music." **–Billy Joel**

K. Monteith, B. Brown, D. Ventura, and T. Martinez, Automatic Generation of Music for Inducing Physiological Responses. *In submission.*

Abstract: *While music is known to have a profound impact on human physiology, the particular physiological responses elicited from defined musical features are not well understood. Because of this, researchers are unable to reliably predict the precise effect of a given piece of music on a given individual. This paper presents a system that is designed to create original musical compositions that elicit particular physiological responses. The experiments described below demonstrate that the music generated by this system is as effective as human-composed music in effecting changes in skin resistance, skin temperature, breathing rate, and heart rate. The system is particularly adept at composing pieces that elicit target responses in individuals who demonstrated predictable responses to training selections.*

Music can have a profound impact on human physiology. It affects how we think, how we feel, and how we relate to others. It captivates and holds our attention, stimulating many areas of the brain. From movie scenes to dance floors, the added sensory input of music makes activities and situations more enjoyable and compelling. One study found that pleasurable music activated the same areas of the brain activated by other euphoric stimuli such as food, sex, or drugs. They highlight the significance of the fact that music would have a similar effect on the brain as "biologically relevant, survival-related stimuli" [Blood and Zatorre, 2001].

Music's impact on human physiology may help explain its long-recognized ability to sway human emotion. It provides not only a medium for expressing a particular emotion, but also the accompanying physiological change to add significance and depth to that emotion. According to the Schachter-Singer theory, emotion is a function of both physiological arousal and cognitive interpretation of that response. The degree of arousal is associated with the degree of emotional response, but it is up to the individual to label that response according to past experience. Schachter and Singer [1962] provide the following illustration in the introduction to one of their study descriptions: "Imagine a man walking alone down a dark alley, a figure with a gun suddenly appears. The perception-cognition 'figure with a gun' in some fashion initiates a state of physiological arousal; this state of arousal is interpreted in terms of knowledge about dark alleys and guns and the state of arousal is labeled 'fear'. Similarly a student who unexpectedly learns that he has made Phi Beta Kappa may experience a state of arousal which he will label 'joy'." Any of the "fight or flight" reactions of the sympathetic nervous system—elevated breathing and heart rate or decreased skin resistance and skin temperature to name a few—can indicate an increase in arousal. While these reactions alone may not be sufficient to label an experienced emotion, they can often be used to indicate the degree of emotional "punch."

Music can also have significant power to calm the body and mind. While relaxation responses such as lowered breathing and heart rate may not be as closely tied with emotional perception and cognition, their elicitation can often have significant therapeutic benefits. One author makes the following observation about maintaining an excessively high state of arousal: "...when individuals chronically overreact to stressful situations, their physiological response system becomes increasingly less flexible. Reduced physiological flexibility is an early indicator of illness, and a chronic physiological overreaction to an external or internal stressor can be viewed as a precipitating, maintaining, or augmenting component of approximately 80% of all illness" [Peper et al., 2008]. Numerous studies have demonstrated the ability of music to induce a relaxation response [e.g White, 1999, Lepage et al., 2001, Khalfa et al., 2002]. Both speed and accuracy of task performance can be enhanced with relaxing music [Allen and Blascovich, 1994].

While there is little question about whether or not music has an effect on humans, predicting the precise effect is more challenging. Most people have a general idea of what types of music make them more energized or relaxed. However, predicting more specific physiological responses is more difficult, particularly if one is trying to make such a prediction about the physiological responses of someone else or about human beings in general. Current research makes few definite claims about the likelihood of specific autonomic responses to specific musical selections. As one researcher explained, "...the large body of literature on physiological responses to music remains characterized by its inconsistencies" [Rickard, 2004]. A few effects do seem to be relatively consistent. For example, one study found that more complex rhythms tended to increase the rate of autonomic functions such as breathing and cardiovascular activity. Silence tended to have the opposite effect–lowering breathing rates and heart rates [Bernardi et al., 2006]. However, even these results only hold true for a majority of individuals. Finding a piece of music that would reliably effect a desired physiological response in a given individual remains a considerable challenge.

Computer-facilitated music may provide some advantages in addressing this challenge. Computers are well-suited to sifting through a large number of both large-scale and fine-grained musical features and of keeping track of which features will most likely have a particular effect. In addition, a human composer might be more biased towards features that would affect his or her own physiology when producing compositions. While a reliance on one's own physiological experiences may be inspiring and helpful in the creative process, when it comes to eliciting physiological responses from others, it may also sometimes result in pieces that are less generalizable. Additionally, once they have "learned" how to do so, computers can generate large quantities of music at virtually no cost in terms of time or effort. A computer would have a much easier time generating a number of different potential compositions to effect a desired result in a given individual until it happened upon the right one. Therefore, the ability of a computer to compose music that elicits a target response at a level even equal to that of humans could have significant benefits.

This paper presents a system capable of generating selections designed to elicit desired physiological responses. It produces original music using statistical models created from a corpus of songs that tend to evoke a targeted response. Preliminary experiments determined likely candidates to populate these musical training corpora. Melodies are constructed using n-gram models representing pitch intervals commonly found in the training corpus for a desired response. Hidden Markov Models are then used to produce harmonies similar to those found in the appropriate corpus. The system also selects the accompaniment pattern and instrumentation for the generated piece based on the various accompaniments and instruments appearing in the target corpus. Neural network evaluators are employed at several points in the creative process to evaluate the generated selections. Data collected in biofeedback experiments with 96 different subjects show that the system is able to generate selections that elicit an average change in a target physiological response with roughly the same ability level as a human performing the same task. The system is particularly effective at eliciting such a response if an individual's response to other musical selections is known.

4.1 Literature Review

This section presents a description of the physiological responses considered in our experiments and provides an overview of research into how these responses are affected by music. It also summarizes work in the area of automatic music generation.

4.1.1 Physiological Responses

Heart rate and breathing rate are two responses commonly monitored in biofeedback experiments. Increases in these measures are arousal responses; decreases indicate relaxation. The typical resting heart rate of an average adult is between 60 and 100 beats per minute [American Heart Association]. Estimates of typical breathing rates are a little more varied, but many sources put the number around 12 to 18 breaths per minutes [Tortora and Anagnostakos, 1990].

Skin temperature is another common biofeedback metric. It reflects the level of blood flow to the underlying tissue. When the body is in a heightened state of arousal, blood is often directed away from peripherals to the internal organs, resulting in a decrease in skin temperature. When the body is more relaxed, blood flow to the hands and feet often increases, and a subsequent raise in skin temperature can be recorded. Typical skin temperature ranges from 18 to 36 degrees Celcius [Peper et al., 2008].

Skin resistance measures can also correspond to the level of arousal. These report the electrical properties of the skin, or more specifically, the activity of the sweat glands. Lower skin resistance corresponds to a higher level of sweat gland activity and a higher level of arousal. Levels range from 100 kOhms for high arousal to 1000 kOhms for low arousal [I-330-C2+ Hardware Guide].

4.1.2 Physiological Responses to Music

A number of studies report conclusive results in experiments using subject-selected music. For example, Allen and Blascovich [1994] studied the effect of music on surgeons performing medical procedures. They reported that autonomic reactivity was significantly lower for subject-selected music than it was to researcher selections. Humans tend to relax more to music that they like. Lepage et al. [2001] reports a similar finding. When patients were allowed to listen to music of their choice before surgery, they required less sedative to achieve the same level of relaxation as patients not listening to music.

There are also a few conclusive studies about how subjects tend to react to researcher-selected music. White [1999] found that heart rate, respiratory rate, and myocardial oxygen demands were lower among patients recovering from myocardial infarctions after they listened to twenty minutes of music, even though the music was experimenter-selected. Khalfa et al. [2002] measured the effects of music on skin resistance. They found that arousal responses were more likely with pieces that the subjects found to communicate happiness or fear. Pieces described as sad or peaceful tended to decrease arousal. As previously mentioned, the work of Bernardi et al. [2006] concluded that more complex rhythms tend to increase the rate of certain autonomic responses and that silence tends to decrease this rate in most people.

4.1.3 Automatic Music Composition

Conklin [2003] summarizes a number of statistical models which can be used for music generation, including random walk, Hidden Markov Models, stochastic sampling, and pattern-based sampling. These techniques can be seen in a number of different works. For example, Chuan and Chew [2007] use Markov chains to harmonize given melody lines, considering melodic notes as observed events and a chord progression as a series of hidden states. Cope [2006] uses statistical models to generate music in a particular style, producing pieces indistinguishable from human-generated compositions.

Delgado et al. [2009] use a rule-based system to generate compositions according to a specified mood. Rutherford and Wiggins [2003] analyze the features that contribute to the emotion of fear in a musical selection and present a system that allows for an input parameter that determines the level of “scariness” in the piece. Oliveira and Cardoso [2007] describe a wide array of features that contribute to emotional content in music and present a system that uses this information to select and transform chunks of music in accordance with a target emotion.

This work employs similar techniques, but rather than focusing on a given mood, it addresses the challenge of generating music designed to elicit a given physiological response.

4.2 Methodology

In order for the music generating system to produce selections that could effectively elicit a desired physiological response, it was first necessary to identify pieces that could be used as training data. This process is described in Section 4.2.1.

Once it is supplied with a training corpus for each target physiological response, the system creates statistical models of the data and uses these models to generate original selections intended to elicit the same responses. The system employs four different components: a Rhythm Generator, a Pitch Generator, a Chord Generator, and an Accompaniment and Instrumentation Planner. The system also makes use of Decision Tree Evaluators to gauge both the pleasantness of the generated melodies and their effectiveness at eliciting a specific physiological response. The functions of the generating and evaluating components of the system are explained in more detail in Section 4.2.2.

In order to evaluate the performance of the system, a human composer was enlisted and asked to complete the same task as the computer: given the information about which songs in the preliminary experiments were most likely to effect changes in the various physiological responses, compose original songs designed to elicit similar physiological changes. Further details about the experiments evaluating the performance of the system are provided in Section 4.2.3.

4.2.1 Training Data Selection

Seventy-two MIDI files were downloaded from the Free MIDI File Database.[1] Themes from movie soundtracks were used due to the wider variety of emotional content available in this genre. The first forty-five seconds of each piece was isolated for use in experiments.

[1] http://themes.mididb.com/movies/

Human Prediction

Since recording physiological responses can be a somewhat time-intensive process, we first selected songs that we predicted as likely to raise or lower the four physiological responses studied (e.g. one corpus consisted of songs that were predicted to be most likely to raise a subject's heart rate, another consisted of songs predicted to lower it). If songs generated from researcher-selected training corpora were just as effective at eliciting desired physiological responses, it would have definite time-saving benefits. This set of songs were also included to provide a more fair comparison between the computer and the human composer. While we furnished our composer with a list of songs that were shown in preliminary experiments to have various effects on physiological responses, these likely informed her compositional choices far less than her own intuition and experiences. Training our system with songs that we selected from our own intuition as likely to have a greater effect on physiological responses allows for the comparison of system-generated and human-composed selections on what might be a more level playing field. Selections chosen by the researchers are shown in Table 4.1.

Experiment Prediction

Next, biofeedback experiments were conducted to more reliably determine effective candidate training pieces. In our preliminary experiments, forty-eight subjects were asked to listen to a number of different training pieces while their heart rate, breathing rate, skin resistance, and skin temperature were monitored. Physiological responses were recorded using the I-330-C2+ biofeedback machine manufactured by J&J Engineering. All were university-enrolled students or professors. Subjects ranged in age from 18 to 52, with the average age being 22. Thirty-four males and 14 females participated.

The seventy-two MIDI selections were split into six groups of twelve selections, and each group of songs was played for eight people.[2] At the beginning of experiments, forty-five

[2] While the song grouping could likely have been randomly assigned without significantly affecting the results, an attempt was made to make the groupings as similar as possible. Acoustic features such as spectral

Table 4.1: Selections predicted to effect various physiological responses

AROUSAL RESPONSES	**RELAXATION RESPONSES**
Raise Breathing Rate	**Lower Breathing Rate**
Eye of the Tiger	Forest Gump
Moral Kombat	Over the Rainbow
What is Love	Casablanca
Axel's Theme	
Raise Heart Rate	**Lower Heart Rate**
Flashdance	Over the Rainbow
Axel's Theme	Forest Gump
Chariots of Fire	Casablanca
Mission Impossible	
Lower Skin Temperature	**Raise Skin Temperature**
Jaws	My Girl
King Kong	Toy Story
Misirlou	Austin Powers
Lower Skin Resistance	**Raise Skin Resistance**
Air Force One	Against All Odds
Star Trek	Superman
Crimson Tide	Good Bad Ugly
Phantom of the Opera	Jurassic Park

seconds of baseline readings were collected. (Subjects were asked to sit quietly and count upwards in their minds during this time in order to achieve neutral results.) Measurements were sampled at one second intervals. For each of the physiological measures, responses were averaged for the duration of baseline readings and for the duration of each of the forty-five second song samples. Then, a z-score was calculated for each of the selections, indicating how many standard deviations the average for a given song varied from the baseline readings.

Responses were then analyzed to determine which selections were most likely to affect a given physiological response. A corpus of training songs comprised of the selections that elicited the largest average change in response was then created for each of the measures studied. These experiment-indicated selections are outlined in Table 4.2.

4.2.2 Automatic Music Generation

Once the training corpora are set, the system develops n-gram models representing melodic movement in a given corpus. Statistics describing the probability of a melody note given a chord and the probability of a chord given the previous chord are collected for each of the various corpora. Information is also gathered about the rhythms, the accompaniment patterns, and the instrumentation present in the songs. This information is then used by the generative and evaluative components of the system, as described in the following sections.

Rhythm Generator

The rhythm for the selection is generated by selecting a phrase from a randomly chosen selection in the training set. The rhythmic phrase is then altered by selecting and modifying a random number of measures. Each measure has a 10% chance of being modified. A

centroid and strongest beat by converting the pieces to MP3 format with winAmp and analyzing them using jAudio software (http://jaudio.sorceforge.net). Symbolic features such as pitch variety and key centeredness were extracted using the jMusic program (http://jmusic.ci.qut.edu.au/). The seventy-two pieces were then split into twelve groups, by making several thousand different random assignments of pieces to groups and selecting the assignment that had the lowest average distance-to-centroid measurements for the twelve clusters based on normalized musical feature vectors.

Table 4.2: Selections that elicited the highest average change in physiological responses in preliminary experiments (shown with average z-score score of effected change)

AROUSAL RESPONSES		**RELAXATION RESPONSES**	
Raise Breathing Rate		**Lower Breathing Rate**	
Mission Impossible	0.90	Bridge Over The River Kwai	-0.58
You're The One That I Want	0.88	Doctor Doolittle	-0.28
Austin Powers	0.84	James Bond	-0.26
Axel's Theme	0.67	Edward Scissorhands	-0.11
Raise Heart Rate		**Lower Heart Rate**	
Batman	1.00	Air Force One	-0.44
Misirlou	0.91	Bridge Over The River Kwai	-0.21
Mission Impossible	0.80	Naked Gun	-0.17
Flashdance	0.78	Beetlejuice	-0.15
Lower Skin Temperature		**Raise Skin Temperature**	
Addams Family	-0.90	Willie Wonka	2.51
1941	-0.64	1492 Conquest of Paradise	2.32
20th Century Fox	-0.51	Air Force One	2.13
That Thing You Do	-0.46	Beetlejuice	2.08
Lower Skin Resistance		**Raise Skin Resistance**	
The Matrix	-0.97	Forrest Gump	0.76
Young Guns	-0.91	Dances With Wolves	0.75
Batman	-0.86	Over The Rainbow	0.73
What is Love	-0.78	Toy Story	0.71

measure is modified by selecting a rhythmic value at random and moving it to the end of the measure.

The system employs a simple AAAA musical form when generating its selections, where each of four successive phrases contains notes with the same rhythm values. Each new rhythmic phrase is evaluated by two decision tree Rhythm Evaluators, described in Section 4.2.2. Generated phrases are only accepted if they are classified positively by both classifiers.

Pitch Generator

Once the rhythm is determined, pitches are selected for the melodic line. Each melody in the training corpus is transposed into a key with no sharps or flats (C major or A minor depending on mode), and then an n-gram model is developed, describing probabilities of melodic progression. When generating a new melody, the system begins with a series of three random notes, selected from a distribution that model which notes are most likely to begin musical selections in the given corpus.

In order to foster cohesion, each phrase is initialized with the same randomly generated three notes. Additional notes in the melodic sequence are randomly selected based on a probability distribution of what note is most likely to follow the given series of three notes. The system generates one hundred possible series of pitches for each rhythmic phrase. As with the rhythmic component, each of the melodies is evaluated by two decision tree Pitch Evaluators, described in Section 4.2.2. Generated melodies are only selected if they are classified positively by both classifiers. In addition, melodies are rejected if they do not end on the tonic pitch unless no such melodies are found among the one hundred possible selections.

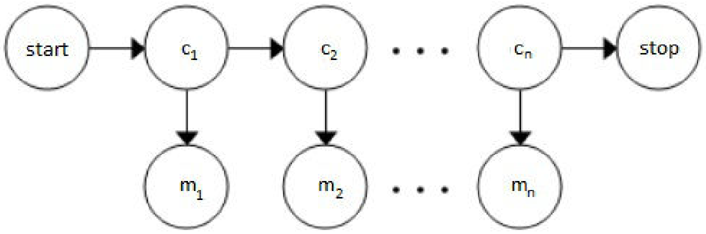


Figure 4.1: Underlying harmony is determined using a Hidden Markov Model, with melody notes considered as observed events and the chord progressions as the underlying state sequence. This requires a knowledge of two conditional probability distributions: the probability of a melody note given a chord, $P(m_i|c_i)$, and the probability of a chord given the previous chord, $P(c_i|c_i - 1)$.

Chord Generator

The underlying harmony is determined using a Hidden Markov Model, with melody notes considered as observed events and the chord progression as the underlying state sequence. The Hidden Markov Model requires two conditional probability distributions: the probability of a set of unique melody notes given a chord and the probability of a chord given the previous chord. The statistics for these probability distributions are gathered from the corpus of music representing the target response. The system then calculates which set of chords is most likely given the melody notes and the two conditional probability distributions. Since many of the songs in the training corpora had only one chord present per measure, the generated harmonizations also follow this assumption. The chord generation process is illustrated in Figure 4.1.

Accompaniment and Instrumentation Planner

The accompaniment pattern for each selection is taken from one of the selections in the training corpus. The system takes as input a measure from a song in the training corpus outlining a characteristic baseline, percussion track, and instrumentation. These input measures act as style files for the computer-generated selections. The system takes the given accompanying measure and transposes it at each measure according to the generated chord

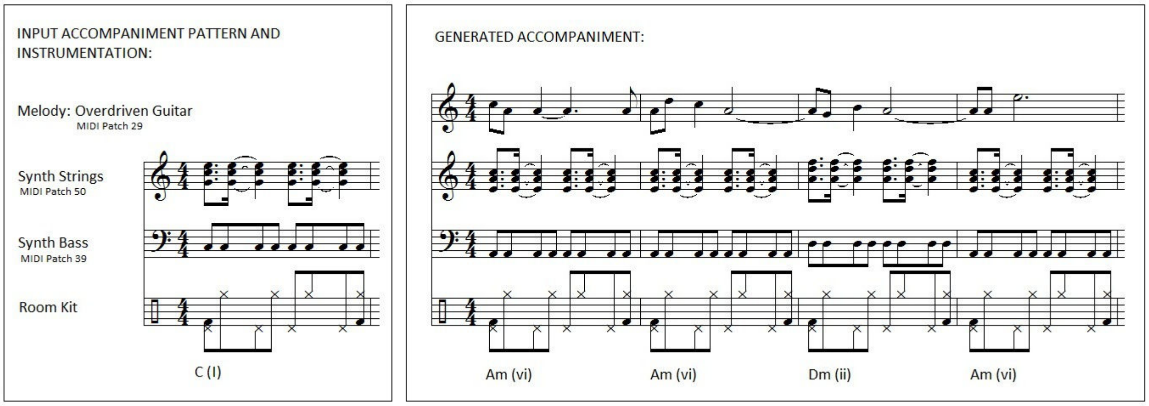


Figure 4.2: An example of how the system fills in an accompaniment pattern for a given melody and set of chords. The input accompaniment pattern and instrumentation are shown on the left. Four bars of the generated melody are shown in the top staff on the right. The generated chords for these measures are Am, Am, Dm, and Am, so the system transposes the input chords according to this pattern (e.g. the repeated 'C' played by the synth bass becomes a repeated 'A' or a repeated 'D').

pattern, producing accompaniments in much the same manner as a pianist selecting a given style on an electronic keyboard. Figure 4.2 provides an example of how the system fills in an accompaniment pattern for a given melody and set of chords. The input accompaniment pattern and instrumentation for *The Matrix*, the top "Experiment Prediction" selection for lowering skin resistance, are shown on the left. Four bars of a melody generated using all four "Experiment Prediction" selections to lower skin resistance (*The Matrix*, *Young Guns*, *Batman*, and *What is Love*) are shown in the top staff on the right. The system-generated chords for these measures are Am, Am, Dm, and Am, so the system transposes the input chords according to this pattern (e.g. the repeated 'C' played by the synth bass becomes a repeated 'A' or a repeated 'D').

Decision Tree Evaluators

Inputs to these classifiers are the default features extracted by the "Phrase Analysis" component of the freely available jMusic software.[3] This component returns a vector of the following twenty-one statistics:

[3] http://sourceforge.net/projects/jmusic/

1. NoteDensity
2. PitchVariety
3. RhythmicVariety
4. ClimaxStrength
5. RestDensity
6. TonalDeviation*
7. KeyCenteredness*
8. PitchRange
9. RhythmRange
10. RepeatedPitchDensity
11. RepeatedRhythmicValueDensity
12. MelodicDirectionStability
13. OverallPitchDirection
14. PitchMovementByTonalStep*
15. Dissonance*
16. LeapCompensation
17. Syncopation
18. RepeatedPitchPatterns
19. RepeatedRhythmPatterns
20. ClimaxPosition
21. ClimaxTonality*

Two values are calculated for each of the starred items, one with a major scale and one with a minor scale provided as input to the "Phrase Analysis" component. Note that the statistics involving pitch are ignored by the decision tree Rhythm Evaluators.

A set of two evaluators is developed for each phase in the evaluation process. The first classifier in each set is trained using analyzed selections in the target corpus as positive training instances and analyzed selections from the other corpora as negative instances (e.g. when generating a selection designed to raise heart rate, statistics about the selections in the training corpus designed to raise heart rate are used as positive inputs to the classifier and statistics about selections in the other seven training corpora are used as negative inputs). This is intended to help the system distinguish selections that elicit specific physiological response.

The second classifier in each set is trained with melodies from all corpora versus thirty-two unevaluated melodies previously generated by the algorithm (e.g. when generating selections using the "Experiment Prediction" training corpora, the system uses statistics from all the selections listed in Table 4.2 as positive inputs and statistics from thirty-two randomly generated, unevaluated melodies as negative inputs). In this way, the system learns to emulate melodies which have already been accepted by human audiences.

Examples of the first decision trees (ones designed to identify features that elicit specific physiological responses) developed at the pitch-assignment phase of the evaluation process are shown in Section 4.3.7.

4.2.3 Evaluation

A second round of biofeedback experiments was conducted to evaluate the generated musical selections. Forty-eight additional subjects participated in the evaluation phase of the experiments. Again, all were university-enrolled students or professors. Subjects ranged in age from 17 to 46, with the average age being 22. Twenty males and 28 females participated.

Physiological responses were recorded for forty songs (sixteen computer-generated selections, sixteen training selections for reference, and eight human-composed selections). To prevent subject fatigue, selections were divided into two groups, one group consisting of pieces targeted to affect breathing and heart rate and one group consisting of pieces targeted to affect skin resistance and skin temperature, and subjects were only asked to listen to one of the groups. Each subject listened to twenty selections; each piece was played for twenty-four people. A Cronbach's alpha coefficient [Cronbach, 1951] was calculated on the responses of subjects in each group to test for inter-rater reliability. Coefficients for the two groups were both $\alpha = 0.99$. (Values over 0.80 are generally considered indicative of a reasonable level of reliability and consequently, a sufficient number of subjects for testing purposes.)

As with the preliminary experiments, baseline readings were collected at the beginning of each recording session. Responses were averaged for the duration of baseline readings and for the duration of each of the selections. Since some individuals were more reactive than others, z-scores are used in analysis instead of absolute changes in measurement.[4]

After listening to each selection, subjects were asked to respond to the following questions (each rated on a scale from 1 to 9):

1. Did you like the selection?
2. How familiar were you with the selection?
3. How musical was the selection?
4. How original was the selection?

4.3 Results

This section provides tables reporting the average z-scores for selections designed to elicit the various target physiological responses. Figures are also provided to illustrate absolute changes in measurement and average z-scores. These provide an idea of the magnitude of the variance in each physiological response.

[4] Recall that z-scores calculate the number of standard deviations an average varies from a given baseline. They are calculated by the formula $z = (x - \mu)/\sigma$, where x is the average for a given selection, μ is the average for baseline, and σ is the standard deviation for readings taken over the duration of the session. Please note that, while z-scores are sometimes used to calculate statistical significance, in this case, these measures are only being used to normalize scores from one individual to the next. Expecting this measure to be above three (a common standard when z-scores are used to calculate significance) would be the equivalent of expecting that out of one hundred songs, only a few had any meaningful effect on physiological responses. A better measure of significance in this case is the Cronbach's alpha value. A high Cronbach's alpha value for a low average z-score indicates that, while a given selection did not tend to elicit a high magnitude change in a response, it was consistent in eliciting a given change for a significant number of subjects.

In each table, the first of the computer-generated selections (corresponding to the first row) was trained using "Human Prediction" training selections. The second was trained using the "Preliminary Experiment" training selections. The third line of the table reports measures recorded for the selection by our enlisted human composer. In most cases, both the computer-generated and human-composed selections were effective at eliciting arousal responses. However, they were less effective at eliciting relaxation responses. This is not surprising considering findings in the literature that music is often more effective at eliciting an arousal response than silence [Bernardi et al., 2006].

Many of the more conclusive studies on the relaxing effects of music deal with subject-selected pieces. Since both the computer-generated and human-composed selections being evaluated are unique to these experiments, subjects would not associate any of them with previous relaxing experiences and consequently experience a relaxation response due to classical conditioning. It would also be difficult for any of the subjects to identify ahead of time which pieces they would find most relaxing. Instead, we look at how subjects responded to the training selections. Each table also reports an adjusted score, calculated by only averaging measurements for individuals for whom the training selections also had the target effect for the measure being considered. While a computer-generated piece may not be able to elicit a particular physiological response in all subjects, this adjusted score allows us to measure whether it elicits a response in a specific group of subjects. (e.g. If it is known that a group of individuals react with a lowered breathing rate to a given song or set of songs, the adjusted score reveals how effective the computer might be in using those training pieces to generate a song that also lowers breathing rate.)

Table 4.3 provides an example of how the adjusted score is calculated. The first row shows individual subject measurements for *Eye of the Tiger*, the tested "Human Prediction" training selection to raise breathing rate (RBR-T1). The second reports scores for the computer-generated selection trained using all the "Human Prediction" training selections for raising breathing rate (RBR-C1). Subjects whose breathing rates were actually raised by the tested training selection (i.e. those with positive scores on the top row) are highlighted, and only these subjects are considered when calculating the adjusted score for the generated selection. The table reports the average for all twenty-four subjects, the adjusted average, and the percentage of the measurements that were included in the adjusted score.

Table 4.3: z-scores for three subjects' responses to a training selection and a computer-generated selection targeted to raise breathing rate. Only the measures for subjects whose breathing rate was raised by the training selection (i.e. those with positive scores on the top row) were considered when calculating the adjusted score for the computer-generated selection.

	1	2	3	4	5	6	7-24	Average	Adjusted Average	% Included in Adj.Avg.
RBR-T1	1.07	-0.01	0.84	0.50	1.99	-1.59	...	0.20		
RBR-C1	1.72	1.34	0.60	-0.41	1.83	-1.48	...	0.46	1.08	50%

4.3.1 Breathing Rate

Figure 4.3 illustrates the range of average absolute changes in breathing rates over the course of a biofeedback session. Responses tended to vary by up to one breath per minute. (Recall that breathing rates tend to vary from 12 to 18 breaths per minute. Considering the small range, an average increase of one breath per minute is non-negligible.) The most significant changes tended towards an increase in breathing rates as compared to baseline.

As shown in Table 4.4, only the computer-generated selection trained with "Experiment Prediction" selections (LBR-C2) was able to successfully lower breathing rate on the average for all subjects. However, the magnitude of the change was small enough that the average change was not significantly different from the other two selections. With the adjusted scores, both computer-generated and human-composed songs (LBR-C1, LBR-C2, and LBR-H) were able to successfully lower breathing rates. Eight individuals–33% of subjects in this group–responded with lowered breathing rates to the tested "Human Prediction" training selection targeted to lower breathing rate (LBR-T1), and six of these individuals also re-

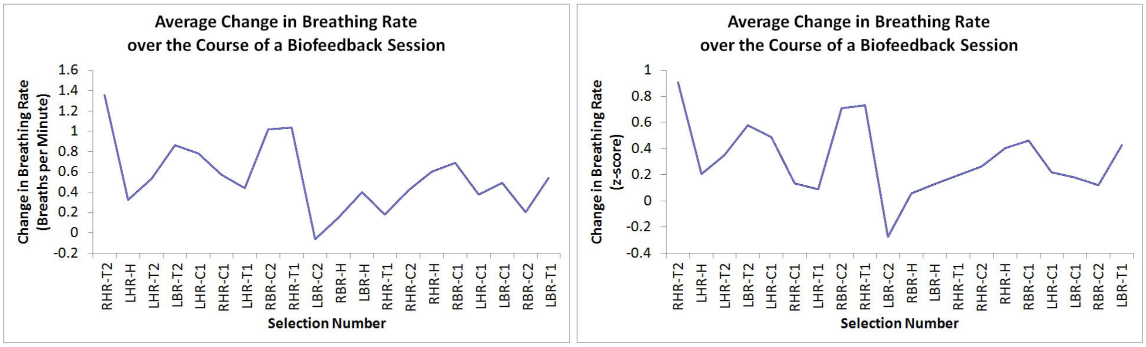


Figure 4.3: Average changes in breathing rates over the course of a biofeedback session.

Table 4.4: Average z-scores of computer and human-generated selections designed to affect breathing rate.

Lower Breathing Rate			
	Overall	Adjusted	
		Average	% Included
Computer-Generated from Human Predictions (LBR-C1)	0.18	-0.32	33%
Computer-Generated from Experiment Predictions (LBR-C2)	-0.27	-1.33	29%
Human-Composed Selection (LBR-H)	0.13	-0.90	29%
Raise Breathing Rate			
	Overall	Adjusted	
		Average	% Included
Computer-Generated from Human Predictions (RBR-C1)	0.46	1.08	50%
Computer-Generated from Experiment Prediction (RBR-C2)	0.71	1.18	46%
Human-Composed Selection (RBR-H)	0.06	0.36	46%

sponded with lowered breathing rate to the computer-generated selection (LBR-C1). Seven individuals–29% of subjects in this group–responded as expected to the top "Experiment Prediction" training selection to lower breathing rate (LBR-T2); four responded similarly to the computer-generated selection (LBR-C2).

The two computer-generated songs designed to raise breathing rate (RBR-C1 and RBR-C2) were able to accomplish this task more effectively than the human-composed song (RBR-H). The 0.71 z-score for the second computer-generated song corresponds to an average increase of over one breath per minute, and the difference in average z-scores between this and the human-generated song was significant at the $p < 0.05$ level. A similar pattern is seen with the adjusted scores. The average difference between the second computer-generated selection and the human-composed song was also significant. Ten of the twelve and nine of the eleven individuals who responded with elevated breathing rate to “Human Prediction” and “Experiment Prediction” training selections targeted to raise breathing rate responded similarly to the computer-generated selections.

Note that the computer-generated selections designed to lower breathing rate are as effective at doing so as the human-composed selections. The computer-generated selections designed to raise breathing rate are performing this task at a level that exceeds that of human performance.

4.3.2 Heart Rate

As shown in Figure 4.4, changes in average heart rate were not quite as pronounced. While individual heart rates could vary by up to fifty beats per minute over the course of a session, the average range for a given individual was only ten beats per minutes. When averaged over all subjects, reactions to songs only varied by a couple of beats per minute.

As shown in Table 4.5, only the human-composed selection (LHR-H) was able to reduce average heart rate, although none of the differences in mean heart rate variation were significant at the $p < 0.05$ level for the three selections. With the adjusted scores, the computer-generated selections (LHR-C1 and LHR-C2) proved more effective at lowering heart rate. For five of the six individuals who responded with lower heart rates to the tested “Human Prediction” training selection and five of the eight individuals whose heart rate lowered to the top “Experiment Prediction” training selection, heart rates were also lowered for the computer-generated songs in these categories.

The computer-generated song trained from the “Experiment Prediction” selections (RHR-C2) was the most effective at raising average heart rate for all subjects. This was followed by the other computer-generated selection (RHR-C1), followed by the human-generated selection (RHR-H). None of the differences between the average changes for these

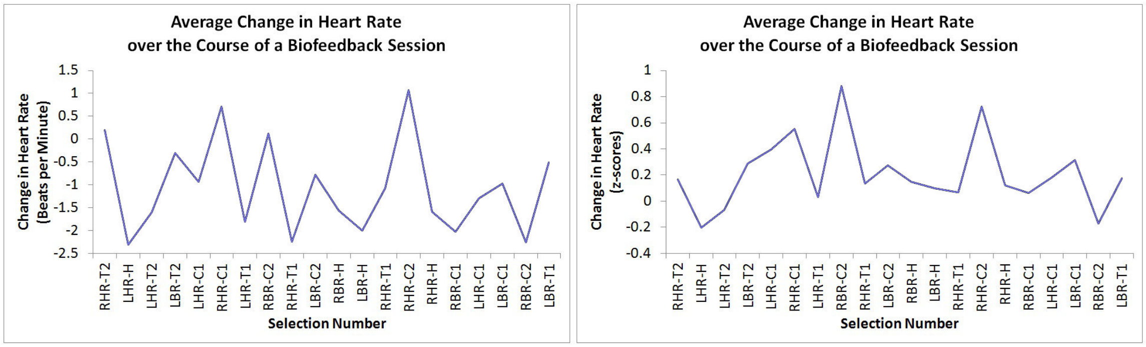


Figure 4.4: Average changes in heart rate over the course of a biofeedback session.

Table 4.5: Average z-scores of computer and human-generated selections designed to affect heart rate.

Lower Heart Rate			
	Overall	Adjusted	
		Average	% Included
Computer-Generated from Human Predictions (LHR-C1)	0.18	-1.10	25%
Computer-Generated from Experiment Predictions (LHR-C2)	0.40	-0.40	33%
Human-Composed Selection (LHR-H)	-0.20	-0.61	33%
Raise Heart Rate			
	Overall	Adjusted	
		Average	% Included
Computer-Generated from Human Predictions (RHR-C1)	0.55	1.31	46%
Computer-Generated from Experiment Predictions(RHR-C2)	0.72	1.09	54%
Human-Composed Selection (RHR-H)	0.12	0.53	54%

three groups were statistically significant. The computer-generated songs were also more effective at raising heart rate using the adjusted score, but not significantly so. Ten of the eleven individuals who responded as expected to the “Human Prediction” training selection and ten of the thirteen individuals who responded as expected to the “Experiment Prediction” training selection to raise heart rate also had their heart rates raised by the computer-generated selections.

As with breathing rate, the computer appears to be addressing the task of composing music that lowers or raises heart rate at a level comparable to that of human performance.

4.3.3 Skin Temperature

Skin temperature tended to rise during the course of the session for most subjects, regardless of the piece of music being played. Figure 4.5 illustrates the average change in skin temperature by selection number for all subjects. Skin temperature tended to rise, on average, by two degrees over the course of a session. Not surprisingly, all selections were better at raising average skin temperature for all subjects than they were at lowering it.

However, when individual subjects did have their skin temperature lowered by a training set selection, they also tended to have their skin temperature lowered by pieces generated from those selections. This was true for three of the four individuals whose skin temperature dropped when listening to a training selection in the "Human Predictions" category (LST-T1) and four of the four individuals whose temperature was lowered by the "Experiment Prediction" selection (LST-T2). The adjusted score for the human-composed selection designed to lower skin temperature was lower than the adjusted scores for the two computer-generated pieces, but the differences were not statistically significant at the $p < 0.05$ level.

Both computer-generated pieces were significantly more effective at raising skin temperature that the human-composed pieces when considering both the regular and the adjusted averages. However, this is almost certainly an artifact of the order in which the pieces were played. (The software used in these experiments did not allow for a randomized order of selection presentation that was unique to each subject.)

While it appears that an effective method of raising skin temperature would simply be composing a piece with sufficient duration, the computer seems as competent at the task as a human. Composing music that lowers skin temperature appears to be a much harder task, but again, these experiments show no statistically significant differences between the performance of the computer and the human.

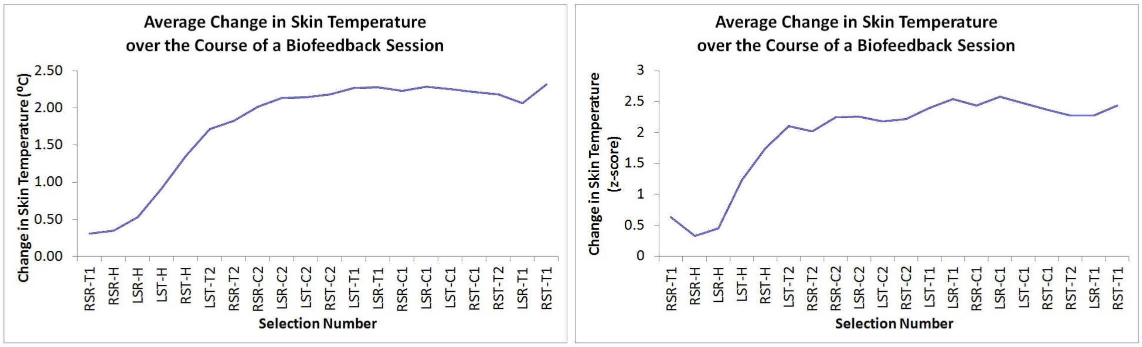


Figure 4.5: Average changes in skin temperature over the course of a biofeedback session.

Table 4.6: Average z-scores of computer and human-generated selections designed to affect skin temperature.

Lower Skin Temperature			
	Overall	Adjusted	
		Average	% Included
Computer-Generated from Human Predictions (LST-C1)	2.47	-1.22	17%
Computer-Generated from Experiment Predictions (LST-C2)	2.18	-1.22	17%
Human-Composed Selection (LST-H)	1.23	-1.84	17%
Raise Skin Temperature			
	Overall	Adjusted	
		Average	% Included
Computer-Generated from Human Predictions (RST-C1)	2.37	3.08	88%
Computer-Generated from Experiment Predictions (RST-C2)	2.22	3.03	83%
Human-Composed Selection (RST-H)	1.75	2.49	83%

4.3.4 Skin Resistance

Table 4.7 reports the average changes in skin resistance to the various musical selections. Most of the selections were likely to elicit an arousal response by lowering skin resistance. There were no significant differences between averages for computer-generated and human-composed songs. However, unlike skin temperature, the effect was not cumulative over the course of the session. Figure 4.6 shows the average change in skin resistance by selection number for all subjects.

For compositions designed to lower skin resistance, there were no significant differences between the two computer-generated selections (LSR-C1 and LSR-C2) and the human-

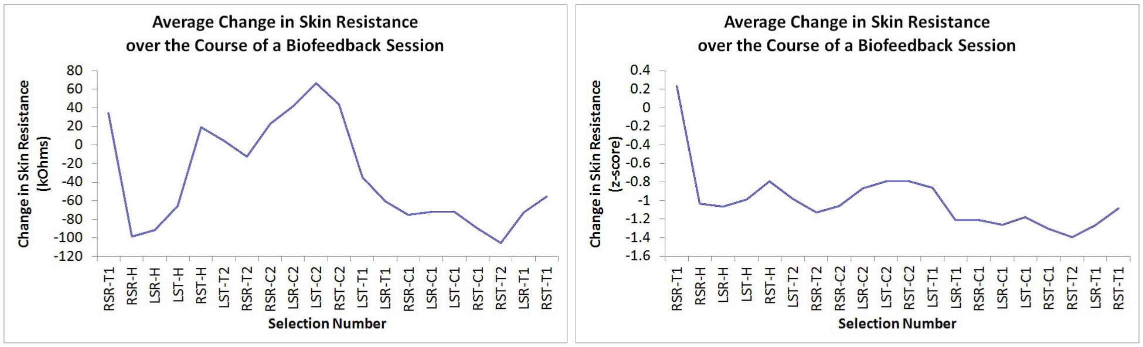


Figure 4.6: Average changes in skin resistance over the course of a biofeedback session.

Table 4.7: Average z-scores of computer and human-generated selections designed to affect skin resistance.

Lower Skin Resistance			
	Overall	Adjusted	
		Average	% Included
Computer-Generated from Human Predictions (LSR-C1)	-1.26	-3.10	58%
Computer-Generated from Experiment Predictions (LSR-C2)	-0.87	-2.48	63%
Human-Composed Selection (LSR-H)	-1.06	-2.00	63%
Raise Skin Resistance			
	Overall	Adjusted	
		Average	% Included
Computer-Generated from Human Predictions (RSR-C1)	-1.21	0.12	63%
Computer-Generated from Experiment Predictions (RSR-C2)	-1.06	2.27	33%
Human-Composed Selection (RSR-H)	-1.03	0.21	33%

generated selection (LSR-H). The "Human Prediction" training selection lowered skin resistance in fourteen individuals, and the "Experiment Prediction" training selection lowered skin resistance in fifteen individuals. With the adjusted scores, both computer-generated selections were more successful at lowering skin resistance than the human-composed song, but differences were only statistically significant between the first computer-generated and the human-composed selection.

There were also no significant differences between the computer-generated selections designed to raise skin resistance (RSR-C1 and RSR-C2) and the human-composed selection (RSR-H). The "Human Prediction" training selection raised skin resistance in fifteen indi-

Table 4.8: Factor analysis summary table

	Loadings			Communalities			
	Factor 1	Factor 2	Factor 3	Factor 1	Factor 2	Factor 3	Percentage
Skin Resistance	-0.058	0.973	0.117	0.003	0.946	0.014	0.963
Skin Temperature	0.917	0.136	-0.117	0.841	0.019	0.014	0.873
Breathing Rate	0.848	-0.308	0.164	0.719	0.095	0.027	0.840
Heart Rate	0.011	0.109	0.985	0.000	0.012	0.971	0.983
	Eigenvalues:			1.563	1.072	1.025	3.660
	Percent of eigenvalues:			0.391	0.268	0.256	0.915

viduals, and the “Experiment Prediction” training selection raised skin resistance in eight individuals. But while the “Experiment Prediction” soundtrack seemed to be less effective at raising skin resistance in terms of number of individuals affected, the subjects for whom it did have the target effect also reacted strongly to the selection generated from all the “Experiment Prediction” soundtracks. The first computer-generated song was barely able to raise average skin resistance, but the second was significantly more effective at raising skin resistance than the human-composed selection at the $p < 0.05$ level.

As with the other measures, the computer is able to generate music that elicits change in skin resistance as effectively or more effectively than a human composition.

4.3.5 Principal Component Plots

In order to provide a visual representation of individual physiological responses to the various selections, three factors were identified through principal component analysis so that responses could be plotted in three dimensions. As shown in Table 4.8, the three rotated factors account for 91.5% of the variance in the data.

Figure 4.7 plots the three rotated factors. As illustrated, higher skin resistance responses appear higher in the positive direction on the Y axis and higher heart rate responses appear farther along the positive Z axis. Both higher breathing rates and higher skin temperature responses appear farther along the X axis.

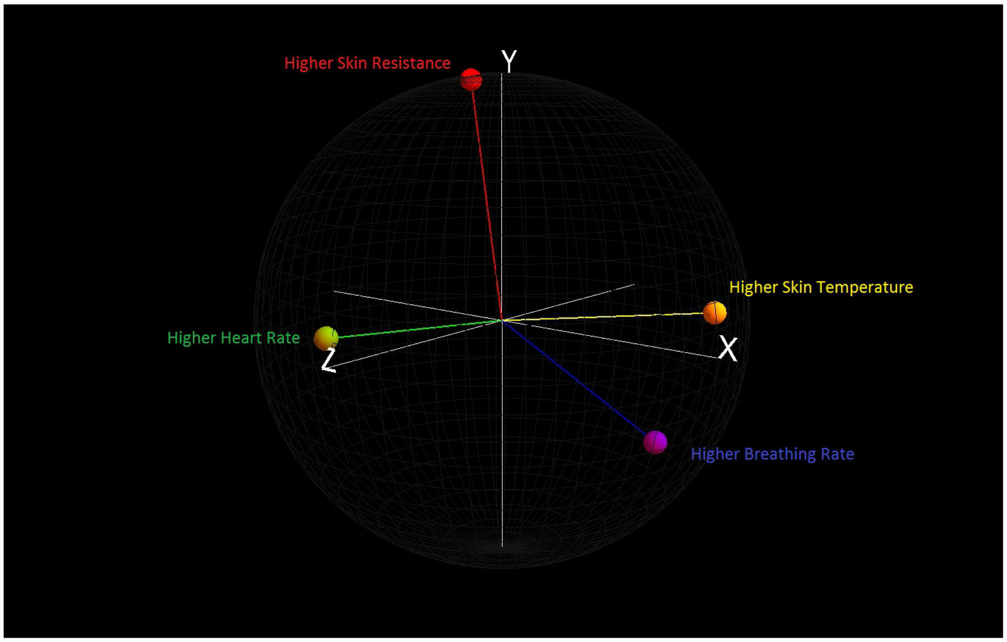


Figure 4.7: Vector Directions

Figure 4.8 highlights responses for pieces either predicted or designed to raise breathing rate. The left image in 4.8 shows factor scores for the responses to *Eye of the Tiger* (RBR-T1) the top "Human Prediction" selection for raising breathing rate. The image on the right shows factors scores for the responses to the computer-generated piece trained from this category of selections (RBR-C1). Colored balls represent the responses of the subjects while listening to a given selection. (Factor scores for a few individuals were eliminated due to missing data resulting from faulty readings.) Pluses represent response factor scores of subjects to the remaining selections. Lines connect responses into clusters as determined by hierarchal agglomerative clustering.

Scores are clustered fairly tightly for the training selection (RBR-T1). While the heart-rate, skin temperature, and skin resistance responses are not as consistent for the computer-generated selection (RBR-C1), it still appears as effective as the training piece at raising breathing rates, as shown by a majority of highlighted response patterns appearing in a positive direction along the X axis.

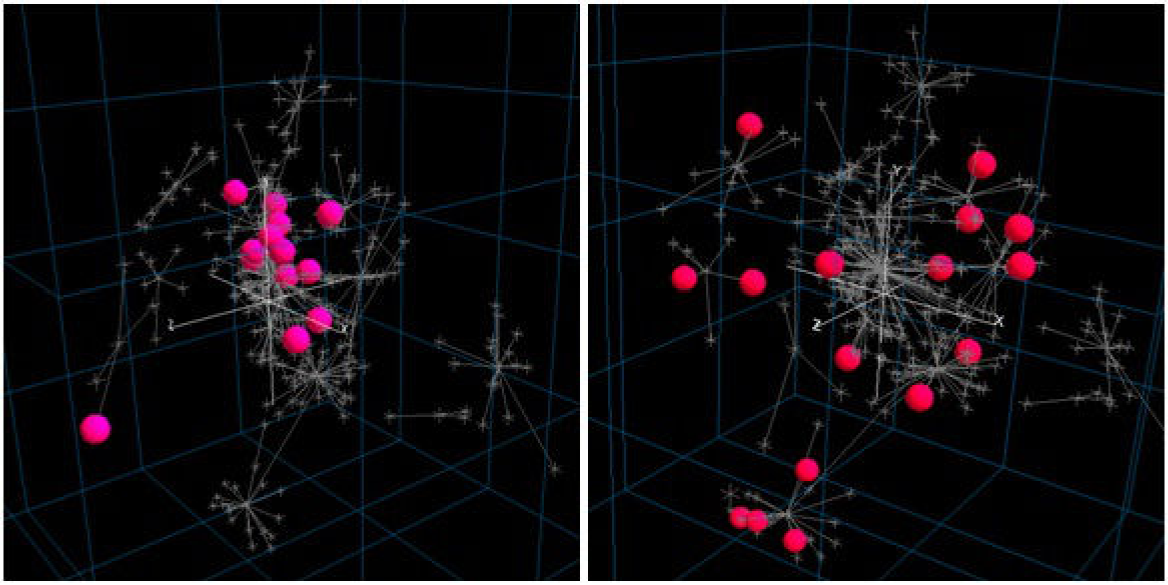


Figure 4.8: Response factor scores for *Eye of the Tiger* (RBR-T1) the top “Human Prediction” selection for raising breathing rate, are highlighted in the image on the left. Response factor scores for the computer-generated piece trained from this category of selections (RBR-C1) are highlighted on the right.

Figure 4.9 shows a similar pattern, with subject responses falling in a similar region for both the top “Experiment Prediction” selection for raising breathing rate and the computer-generated piece trained from this category of selections. Again, selections tend to lie in a positive direction along the X axis, indicating a tendency to raise breathing rate.

Figures highlighting response patterns for all the computer-generated selections are provided in the Appendix. These figures illustrate that selections varied in their effectiveness at raising and lowering a target physiological responses. But they also illustrate how computer-generated pieces could be remarkably effective at eliciting the same physiological responses as some of the selections from which they were trained (e.g. see Figure 4.18). If an individual’s response pattern is known, our system appears to be quite adept at generating unique new selections that also produce these responses.

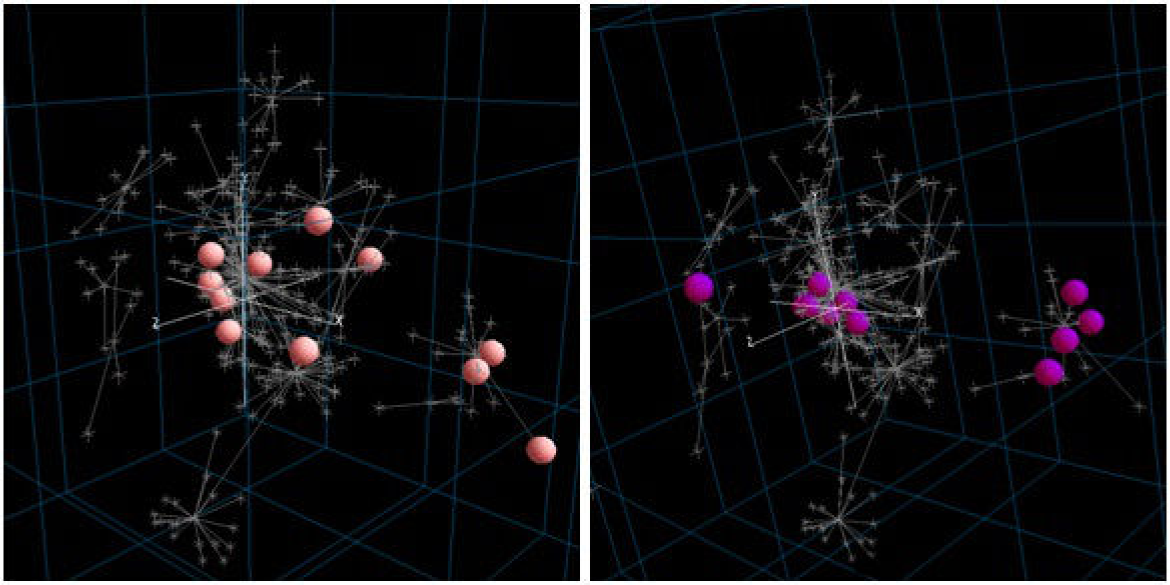


Figure 4.9: Response factor scores for *Mission Impossible* (RBR-T2) the top "Experiment Prediction" selection for raising breathing rate, are highlighted in the image on the left. Response factor scores for the computer-generated piece trained from this category of selections (RBR-C2) are highlighted on the right.

4.3.6 Subjective Responses

Average responses to the subjective questions asked after each selection are shown in Table 4.9. Not surprisingly, the initial training selections and the human-composed selections received higher rating for likability and musicality. However, the computer-generated selections received slightly higher ratings for originality and significantly lower ratings for familiarity than the training selections and human-composed selections–evidence to suggest that the computer is producing genuinely original compositions and not borrowing too heavily from training data.

As shown in Table 4.10, there was no correlation between subjective responses and physiological changes. While for some individuals, liking a song might result in a more dramatic increase or decrease in a given physiological response, this does not appear to be the case overall.

Table 4.9: Average results to subjective questions (Responses were measured on a scale of 1 to 9)

Did you like the selection?	
Training Selections	5.83
Computer-Generated Selections	3.97
Human-Composed Selections	5.56
How familiar was the selection?	
Training Selections	5.53
Computer-Generated Selections	2.17
Human-Composed Selections	3.01
How musical was the selection?	
Training Selections	5.35
Computer-Generated Selections	3.88
Human-Composed Selections	5.12
How original was the selection?	
Training Selections	6.36
Computer-Generated Selections	6.97
Human-Composed Selections	6.70

Table 4.10: Correlations between subjective responses and physiological measures

	Like	Familiar	Musical	Original
Breathing Rate	0.02	0.03	0.03	-0.04
Heart Rate	0.03	-0.01	0.04	0.09
Skin Temperature	0.04	0.04	0.00	-0.06
Skin Resistance	-0.03	-0.05	-0.02	-0.03

4.3.7 Musical Features

Figure 4.10 outlines the characteristics identified by the evaluating decision trees as being responsible for various physiological responses. For example, if a melody had a ClimaxPosition measure of 0.67 and a Dissonance measure greater than 0.01 or PitchMovementByTonalStep measure greater than 0.63, it was classified as a good candidate for raising heart rate. Informally, pieces that raised heart rates tended to have more dissonance and more scale-wise movement. Pieces that lowered heart rate, on the other hand, tended to have less rhythmic variety (perhaps contributing to more flowing rhythms) and a stronger climax.

Raise Heart Rate	**Lower Heart Rate**
ClimaxPosition <= 0.67	ClimaxTonality <= 0: No
— Dissonance <= 0.01	ClimaxTonality > 0
— — PitchMovementByTonalStep <= 0.63: No	— RhythmicVariety <= 0
— — PitchMovementByTonalStep > 0.63: Yes	— — ClimaxStrength <= 0.33: No
— Dissonance > 0.01: Yes	— — ClimaxStrength > 0.33: Yes
ClimaxPosition > 0.67: No	— RhythmicVariety > 0: No
Raise Breathing Rate	**Lower Breathing Rate**
RhythmicVariety <= 0: No	RhythmicVariety <= 0.06
RhythmicVariety > 0	— Syncopation <= 0.18: No (20.0)
— ClimaxTonality <= 0: Yes	— Syncopation > 0.18
— ClimaxTonality > 0	— — OverallPitchDirection <= 0.49: No
— — ClimaxStrength <= 0.25: Yes	— — OverallPitchDirection > 0.49: Yes
— — ClimaxStrength > 0.25: No	RhythmicVariety > 0.06: Yes
Lower Skin Temperature	**Raise Skin Temperature**
MelodicDirectionStability <= 0.41: No	ClimaxTonality2 <= 0.5: No
MelodicDirectionStability > 0.41	ClimaxTonality2 > 0.5
— ClimaxTonality <= 0: Yes	— ClimaxStrength <= 0.25: No
— ClimaxTonality > 0	— ClimaxStrength > 0.25
— — PitchRange <= 0.67: No	— — PitchMovementByTonalStep2 <= 0.15: No
— — PitchRange > 0.67: Yes	— — PitchMovementByTonalStep2 > 0.15: Yes
Lower Skin Resistance	**Raise Skin Resistance**
PitchVariety <= 0.11	RhythmicRange <= 0.97: No
— MelodicDirectionStability <= 0.29: Yes	RhythmicRange > 0.97
— MelodicDirectionStability > 0.29: No	— RhythmicVariety <= 0
PitchVariety > 0.11: No	— — MelodicDirectionStability <= 0.29: No
	— — MelodicDirectionStability > 0.29: Yes
	— RhythmicVariety > 0: No

Figure 4.10: Decision tree models of musical characteristics contributing to changes in various physiological measures

Melodies that tended to raise breathing rate tended to higher rhythmic variety and either a non-tonal climax note or lower climax strength. Somewhat surprisingly, melodies that lowered breathing rate also tended to have higher rhythmic variety, but also some syncopation and a tendency to upward pitch direction.

Features contributing to a lowered skin temperature response included stability of melodic direction and a non-tonal climax. In other words, upward movement towards a climax that involved a non-tonal suspension note were arousing. A greater pitch range also contributed to lowered skin temperature. Pitch movement by minor tonal step leading to a strong climax tended to contribute to raised skin temperature.

Melodies that tended to lower skin resistance had lower pitch variety and less stability of melodic direction; some of these arousing melodies tended to bounce back and forth between notes. Melodies that raised skin resistance had a greater stability of melodic direction, as well as less rhythmic variety and range.

4.4 Discussion

Tables 4.11 and 4.12 provide a summary of how effective we were at eliciting a change in physiological responses in various situations.

None of the selections were able to lower average skin temperature, but both computer-generated and human-composed selections designed to elicit the other arousal responses (raised breathing rate, raised heart rate, and lowered skin resistance) were, on average, able to do so successfully. In the case of breathing rate, one of the human generated songs was able to raise breathing rate more effectively than the human-composed song at a level that was significant. The computer is performing the task of eliciting arousal responses at a level equal to or greater than human ability.

When considering only subjects who responded as expected to the training selections, both the computer-generated and human composed songs were successful at eliciting an average arousal response for all of the measures studied. For breathing rate and skin resistance, the differences between one of the computer-generated selections and the human-composed selection were significant, the computer-generated one being more effective at eliciting the target response.

Eliciting relaxation responses proved a little more challenging for both the computer-generated and human-composed selections. All were able to raise skin temperature, but none were able to raise skin resistance. Only one of the computer-generated selections was able to lower heart rate, and only the human-composed selection was able to lower breathing rate. Differences between the computer-generated and human-composed songs were insignificant–

Table 4.11: Summary of success of eliciting an arousal response to musical stimuli from various sources

	Raise Breathing Rate	Raise Heart Rate	Lower Skin Temperature	Lower Skin Resistance
Computer-Generated from Human Predictions	✓	✓	✗	✓
Computer-Generated from Preliminary Experiment Predictions	✓	✓	✗	✓
Human-Composed Selection	✓	✓	✗	✓
Computer-Generated from Human Predictions (Adjusted)	✓	✓	✓	✓
Computer-Generated from Preliminary Experiment Predictions (Adjusted)	✓	✓	✓	✓
Human-Composed Selection (Adjusted)	✓	✓	✓	✓

the computer was able to perform at a level equal to that of human performance at the task of generating songs that elicit relaxation responses in all individuals.

When considering adjusted scores, both the computer-generated and human composed selections were able to elicit all target relaxation responses. In the case of skin resistance, one of the computer-generated songs was significantly better at raising average response. Again, the computer is performing at a level at or above that of human performance.

Overall, the system proves itself able to generate songs that elicit target physiological responses with similar effectiveness to songs generated by a human composer. Both still require information about a given individual's physiological responses in order to generate a new piece that also reliably elicits those responses in many categories. However, given the variability of human biofeedback responses, the ability to consistently effect targeted physiological responses under any conditions can be viewed as fairly impressive.

There appears to be little difference between selections generated from the "Human Prediction" training corpora and ones generated using the "Experiment Prediction" training

Table 4.12: Summary of success of eliciting a relaxation response to musical stimuli from various sources

	Lower Breathing Rate	Lower Heart Rate	Raise Skin Temperature	Raise Skin Resistance
Computer-Generated from Human Predictions	✗	✗	✓	✗
Computer-Generated from Preliminary Experiment Predictions	✗	✓	✓	✗
Human-Composed Selection	✓	✗	✓	✗
Computer-Generated from Human Predictions (Adjusted)	✓	✓	✓	✓
Computer-Generated from Preliminary Experiment Predictions (Adjusted)	✓	✓	✓	✓
Human-Composed Selection (Adjusted)	✓	✓	✓	✓

corpora in terms of effectiveness in eliciting particular physiological responses. Subjects who responded as expected to the training data (e.g. subjects whose breathing rate decreased while listening to a selection in the corpus of training songs predicted to lower breathing rate) tended to also respond as expected to a piece generated from that data, regardless of the process used in training corpus compilation.

In these experiments, we borrowed heavily from the accompaniment pattern of training songs. Three subjects out of the forty-eight who listened to any computer-generated selections asked if one of the computer-generated songs was a repeat of a previous song. One was actually able to identify both the source of the accompaniment and elements of the melody. ("It sounds a little like *Axel's Theme* with a different rhythm layered over a *Mission Impossible* baseline.") More generic results could come from using pre-programmed style files or loop libraries. However, the principles demonstrated here would likely still apply. We predict that subjects who responded with a given physiological response to a given style file would be much more likely to respond similarly to a computer-generated piece employing that particular style.

Not surprisingly, the human-composed selections received higher average ratings for likability, but there were a number of positive anecdotal responses to some of the computer-generated selections. One subject mentioned that the first computer-generated piece designed to raise heart rate (RHR-C1) would be well-suited to accompany an exciting movie scene. Another thought the first computer-generated piece designed to raise skin temperature (RST-C1) would make an excellent video game soundtrack. Hearing what she thought were eastern influences, another subject thought the second computer-generated piece to lower skin resistance (LSR-C2) would be good ambient music for an oriental restaurant. While the musicality of the computer-generated selections could still be improved, most of the selections were by no means unpleasant. Further work may also include refinements to the evaluating decision trees and possibly other aspects of the system's generative process in order to allow it to produce more musical and pleasing selections.

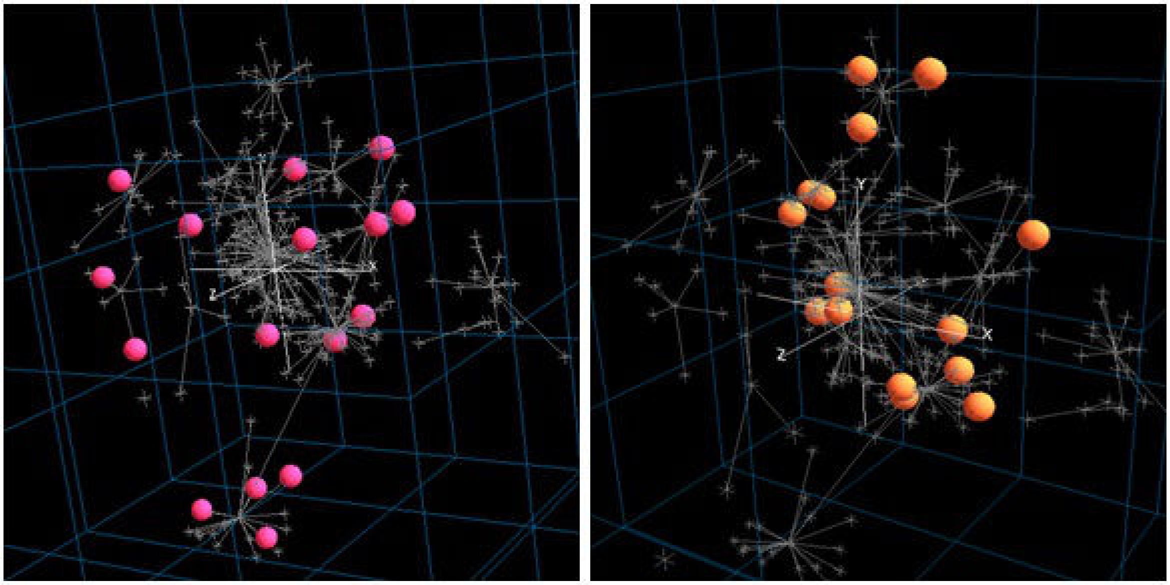


Figure 4.11: Response factor scores for LBR-T1 are highlighted on the left. Response factor scores for LBR-C1 are highlighted on the right.

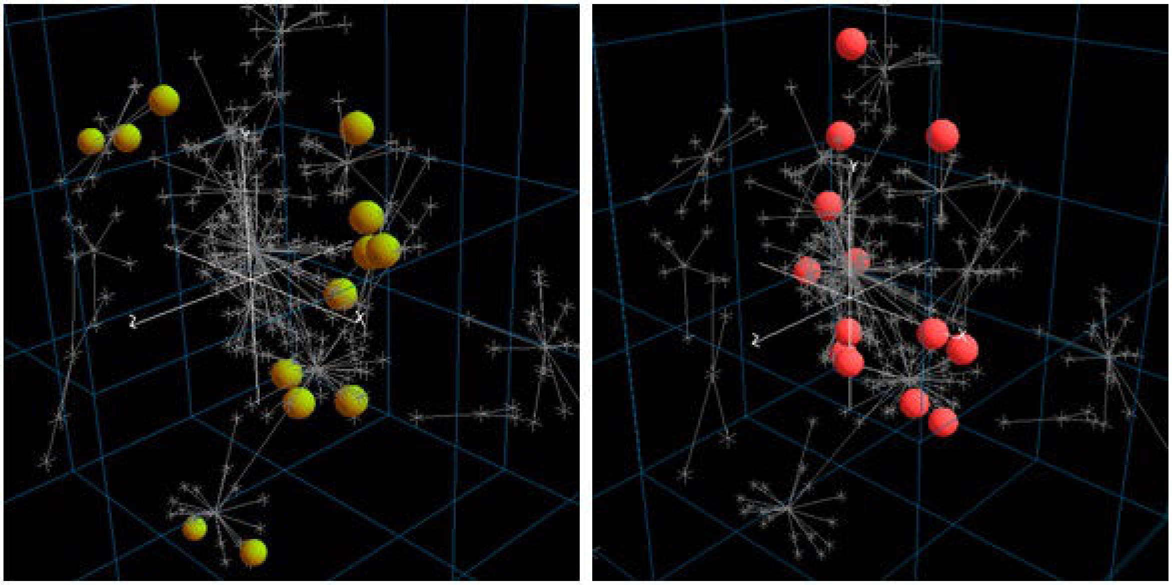


Figure 4.12: Response factor scores for LBR-T2 are highlighted on the left. Response factor scores for LBR-C2 are highlighted on the right.

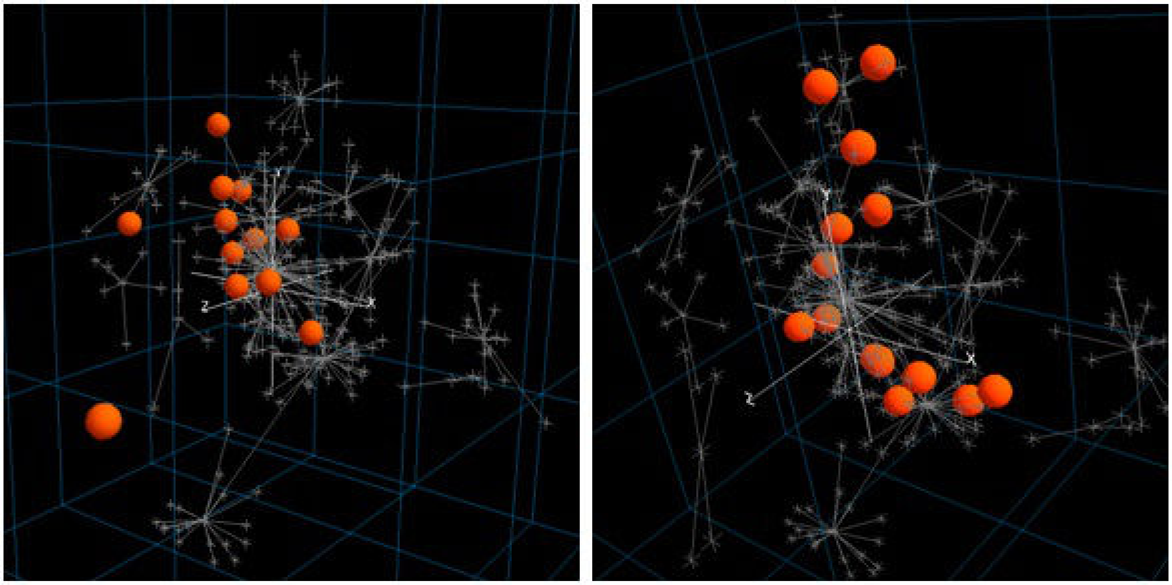


Figure 4.13: Response factor scores for RHR-T1 are highlighted on the left. Response factor scores for RHR-C1 are highlighted on the right.

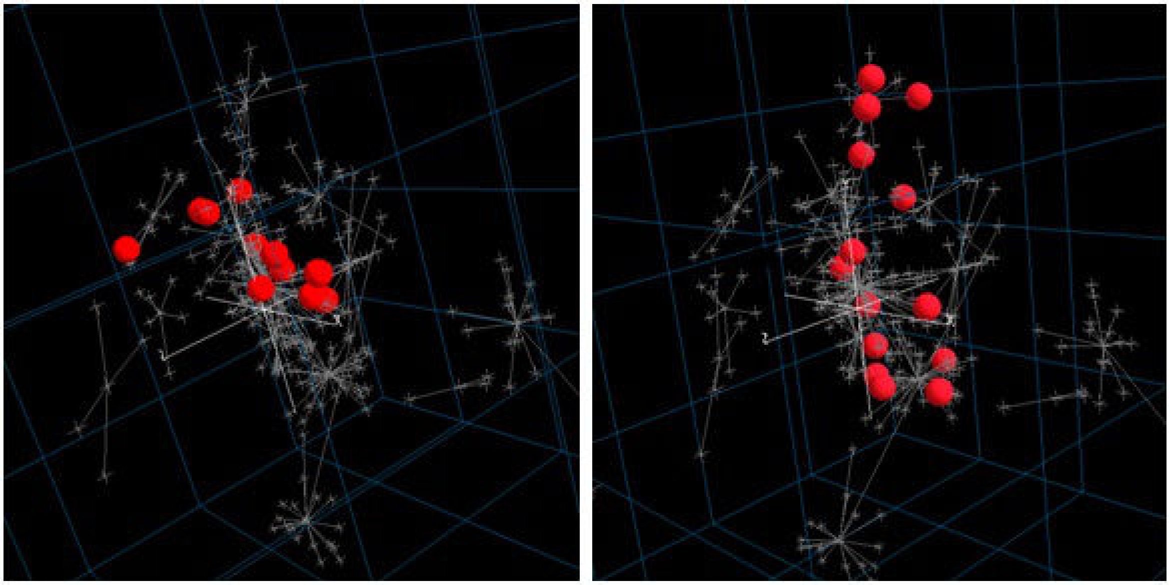


Figure 4.14: Response factor scores for RHR-T2 are highlighted on the left. Response factor scores for RHR-C2 are highlighted on the right.

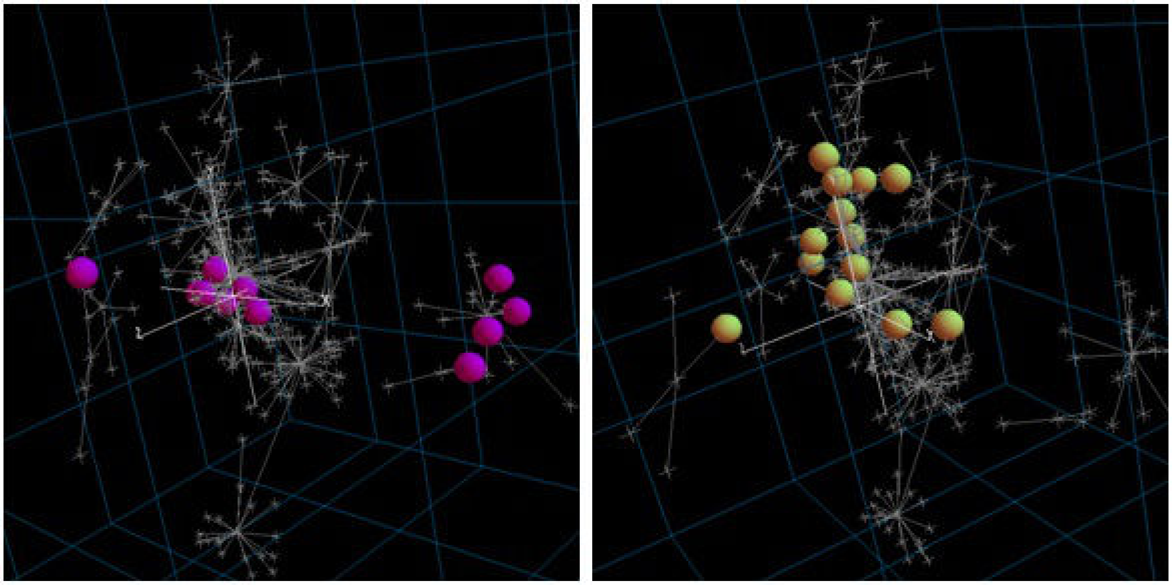


Figure 4.15: Response factor scores for LHR-T1 are highlighted on the left. Response factor scores for LHR-C1 are highlighted on the right.

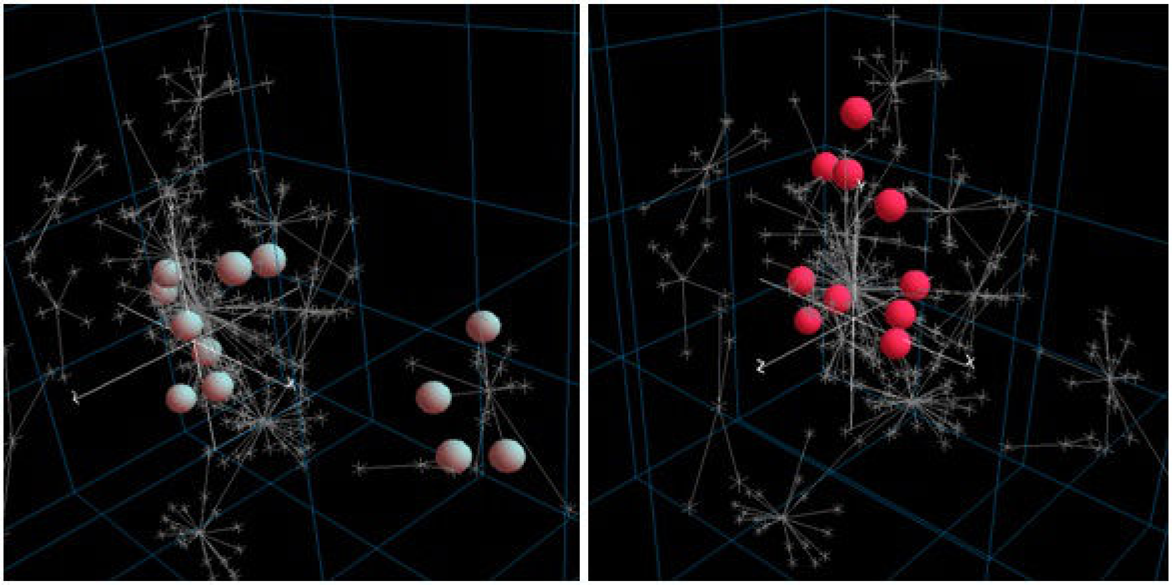


Figure 4.16: Response factor scores for LHR-T2 are highlighted on the left. Response factor scores for LHR-C2 are highlighted on the right.

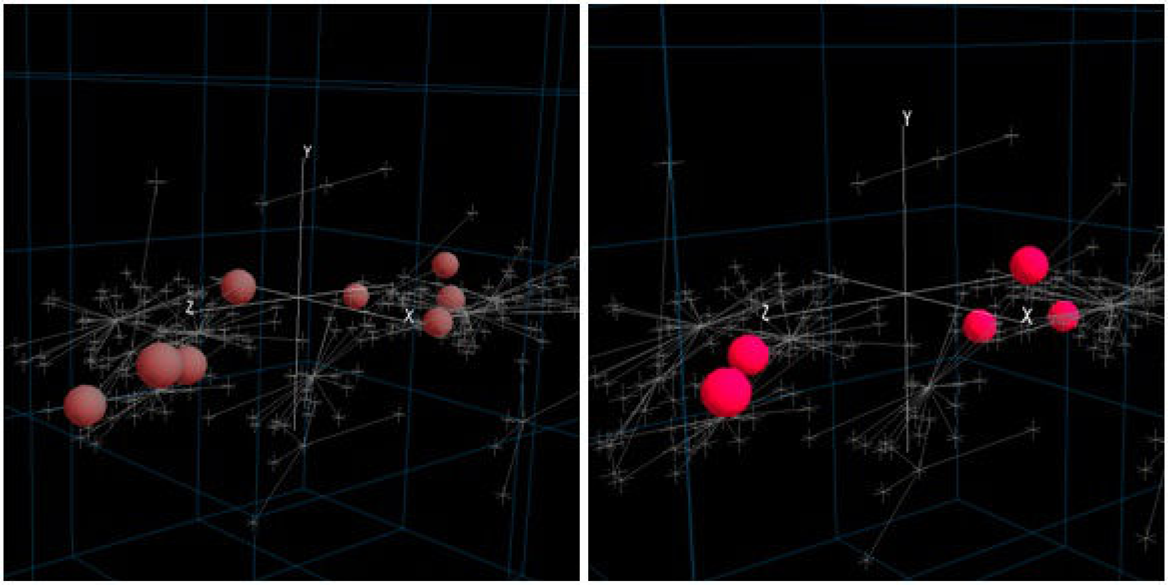


Figure 4.17: Response factor scores for LST-T1 are highlighted on the left. Response factor scores for LST-C1 are highlighted on the right.

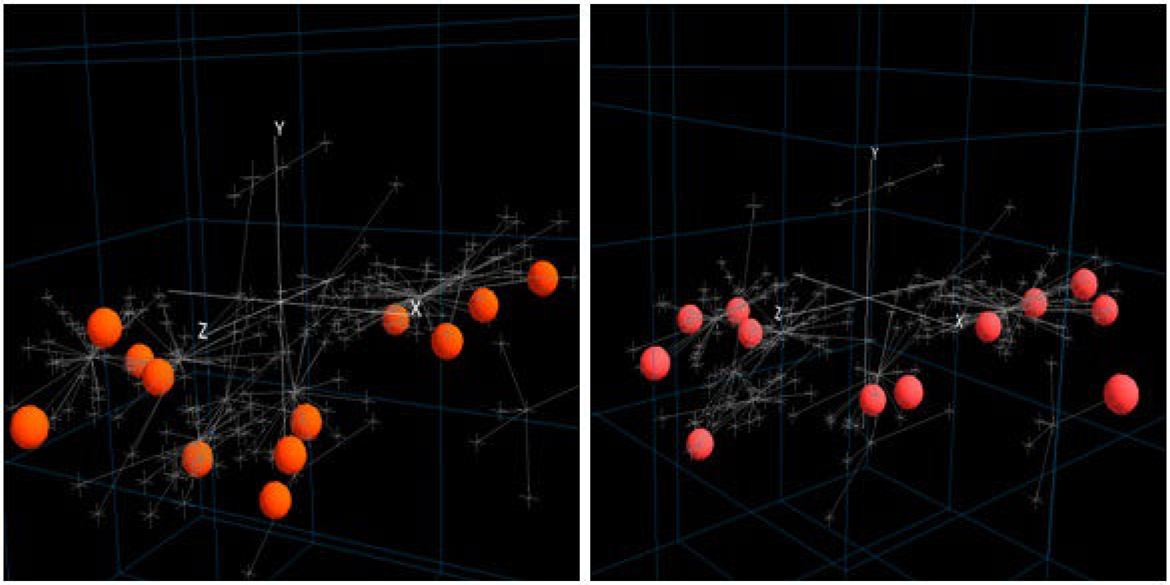


Figure 4.18: Response factor scores for LST-T2 are highlighted on the left. Response factor scores for LST-C2 are highlighted on the right.

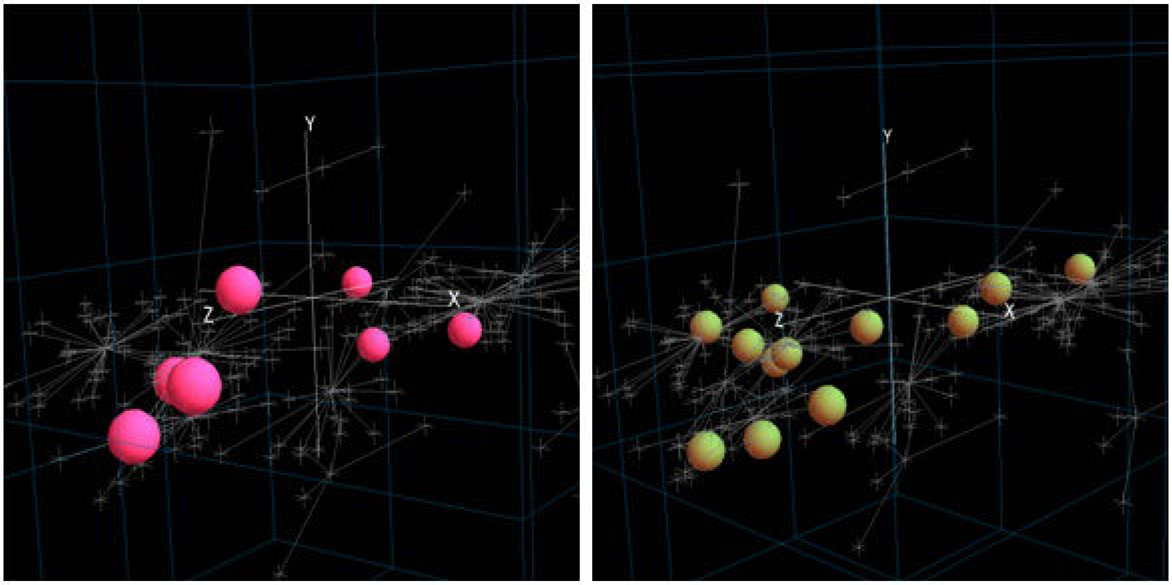


Figure 4.19: Response factor scores for RST-T1 are highlighted on the left. Response factor scores for RST-C1 are highlighted on the right.

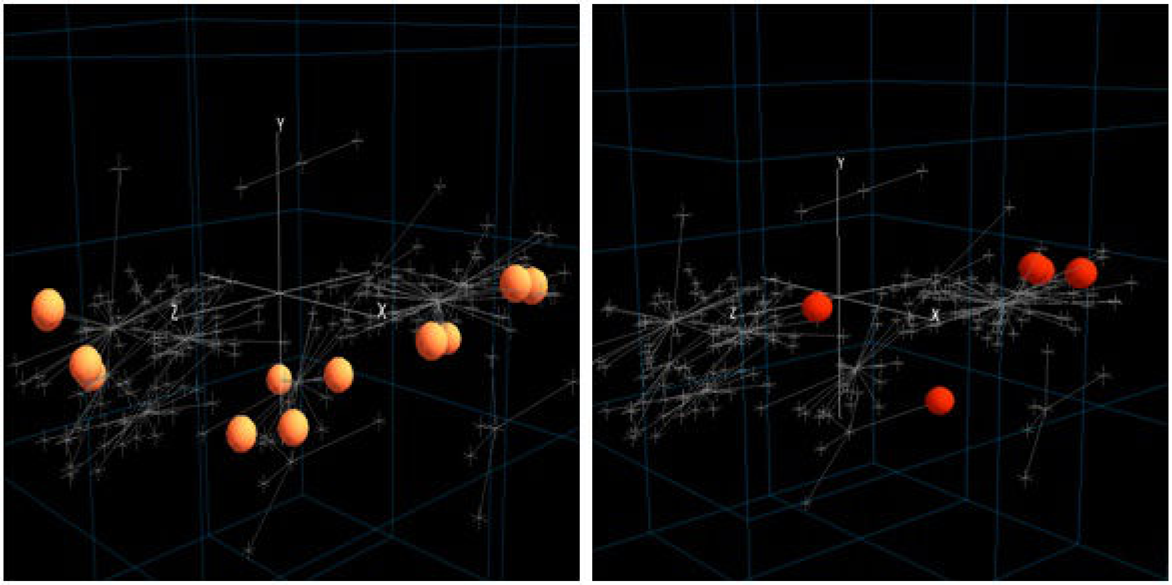


Figure 4.20: Response factor scores for RST-T2 are highlighted on the left. Response factor scores for RST-C2 are highlighted on the right.

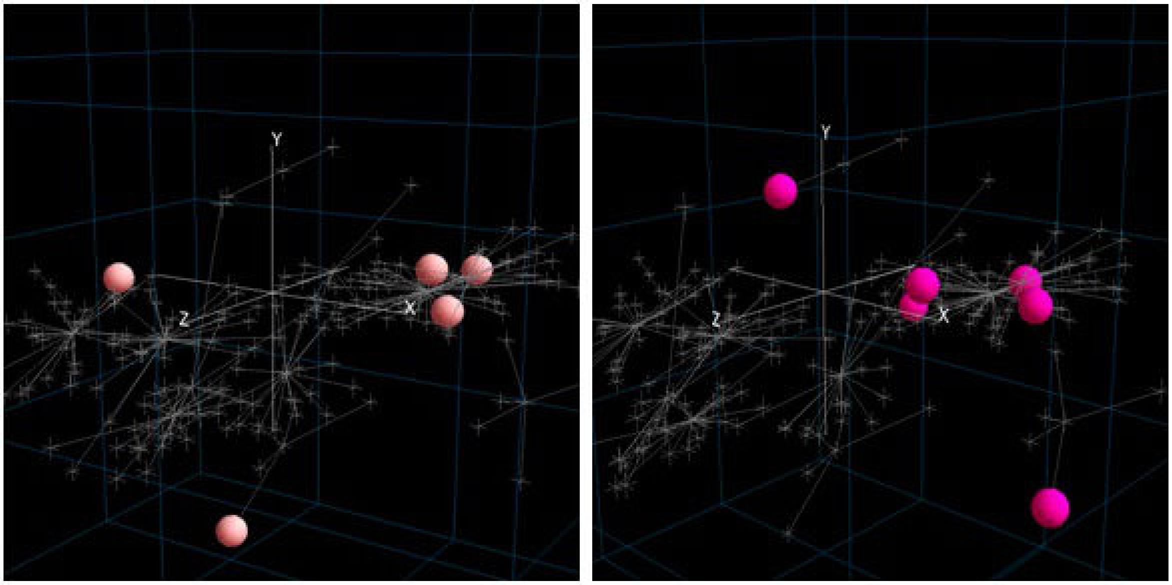


Figure 4.21: Response factor scores for LSR-T1 are highlighted on the left. Response factor scores for LSR-C1 are highlighted on the right.

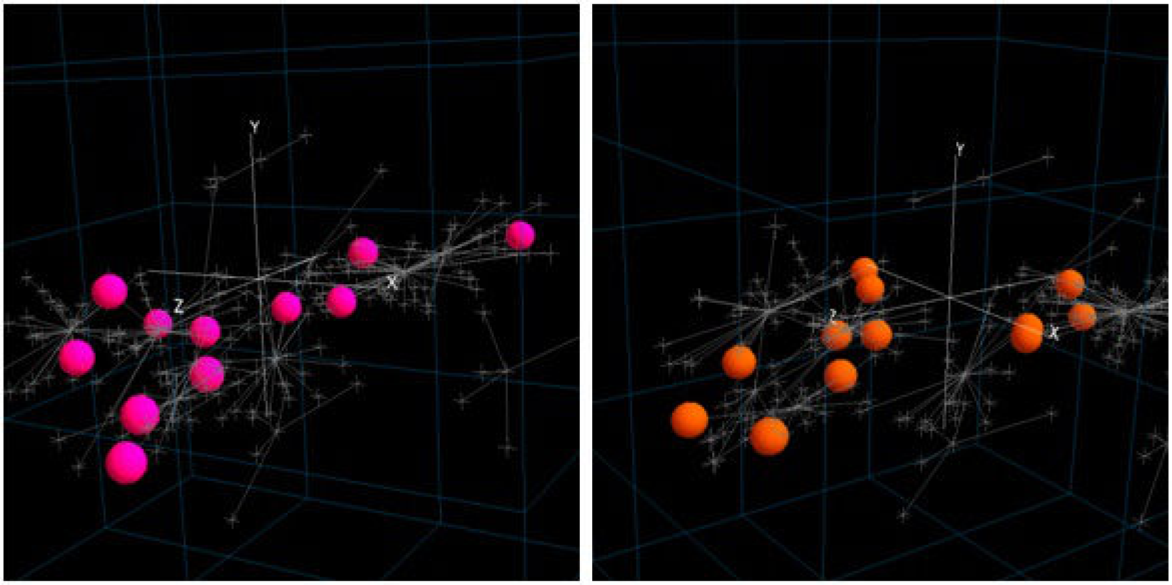


Figure 4.22: Response factor scores for LSR-T2 are highlighted on the left. Response factor scores for LSR-C2 are highlighted on the right.

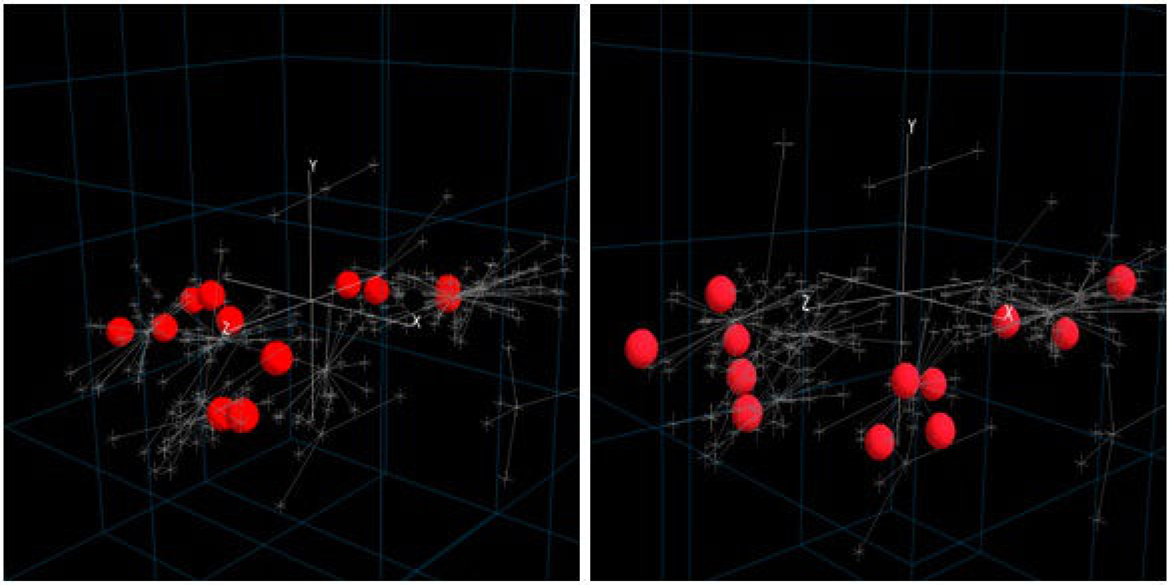


Figure 4.23: Response factor scores for RSR-T1 are highlighted on the left. Response factor scores for RSR-C1 are highlighted on the right.

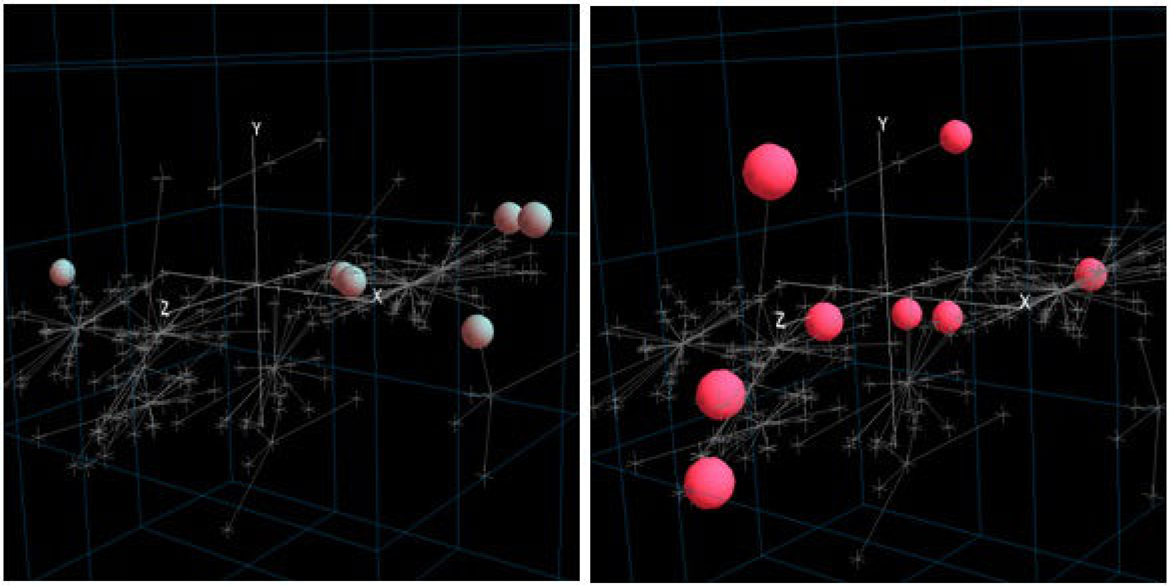


Figure 4.24: Response factor scores for RSR-T2 are highlighted on the left. Response factor scores for RSR-C2 are highlighted on the right.

Chapter 5

Automatic Generation of Emotionally-Targeted Soundtracks

"Music is nothing else but wild sounds civilized into time and tune."
–Thomas Fuller

Published in shortened form as: K. Monteith, V. Francisco, T. Martinez, P. Gervás, and D. Ventura. Automatic Generation of Emotionally-Targeted Soundtracks. In *Proceedings of the Second International Conference on Computational Creativity*, pages 60-62, 2011.

Abstract: *Music can be used both to direct and enhance the impact a story can have on its listeners. This work makes use of two creative systems to provide emotionally-targeted musical accompaniment for stories. One system assigns emotional labels to text, and the other generates original musical compositions with targeted emotional content. We use these two programs to generate music to accompany audio readings of fairy tales. Results show that music with targeted emotional content makes the stories significantly more enjoyable to listen to and increases listener perception of emotion in the text.*

5.1 Introduction

Music has long been an integral aspect of storytelling in various forms of media. In the era of silent films, music was used to drown out the extraneous noise of the projector [Cohen, 2001], but even when spoken dialog was added and projectors were quieted, music remained an integral part of cinematography [Johnson, l969]. Research indicates that soundtracks can be very effective in increasing or manipulating the affective impact of a story. For example, Thayer and Levenson [l983] found that musical soundtracks added to a film about industrial safety could be used to both increase and decrease viewers' electrodermal responses depending on the type of music used. Bullerjahn and Guldenring [1994] similarly found that music could be used both to polarize the emotional response and impact plot interpretation. Marshall and Cohen [l988] noted significant differences in viewer interpretation of characters in a film depending on the type of accompanying music. Boltz [2004] found that, if musical soundtracks were congruent with a story, it made both the music and the story more memorable. As she explains in her research, music appears to function as a schema, or framework in which to interpret events. It "clarifies the characters' temperaments and their relationships to one another, as well as clarifying their actions and underlying motivations" and "guides attending toward those aspects of a film that are consistent with this framework, and thereby determines which elements are remembered best." Music can even affect the behavior of individuals after hearing a story. For example, Brownell [2002] found that, in several cases, a sung version of a story was more effective at reducing an undesirable target behavior than a read version of the story.

Clearly, music can have a significant impact on the telling of a story, and specially targeted music can both direct and enhance this impact. This paper investigates the possibility of automatically generating emotionally targeted music to accompany the reading of fairy tales. The task requires both the ability to label text with the appropriate emotional tags and to generate music that expresses those emotions. We review a system that is able to take a list of possible emotional tags and assign these labels to a given piece of text. We then review a system that is able to generate original music with desired emotional content. These two systems are combined to address the task of automatically generating music to accompany audio recordings of fairy tales. Results show that emotionally targeted music makes stories significantly more enjoyable and causes them to have a greater emotional impact than music that is generated without regard to the emotions inherent in the text.

5.2 Related Work

Programmers and researchers have often attempted to endow machines with some form of intelligence. In some cases, the end goal of this is purely practical; a machine with the capacity to learn could provide a multitude of useful and resource-saving tasks. But in other cases, the goal is simply to make machines behave in a more creative or more "human" manner. As one author explains, "Looked at in one way, ours is a history of self-imitation...We are ten times more fascinated by clockwork imitations than by real human beings performing the same task" [McCorduck, 2004]. One major area of human creativity involves the production of music. Wiggins [2006] states that, "...musical behavior is a uniquely human trait (notwithstanding our anthropomorphic tendencies in terminology such as bird and whale-'song', which are in fact much more like language than music); further, it is also ubiquitously human: there is no known human society which does not exhibit musical behaviour in some form." Naturally, many computer science researchers have turned their attention to musical computation tasks. Researchers have attempted to classify music, measure musical similarity, and predict the musical preferences of users [Chai and Vercoe, 2001, Li and Ogihara, 2004, McKay and Fujinaga, 2004, Pampalk et al., 2005]. Others have investigated the ability to search through, annotate, and identify audio files [Dannenberg et al., 2003, Dickerson and Ventura, 2009]. More directly in the realm of computational creativity, researchers have developed systems that can automatically arrange and compose music [de la Puente et al., 2002, Conklin, 2003, Allan and Williams, 2005]. This paper extends the task of automatic musical composition to address the challenge of composing music to accompany specific text.

Computing that possesses some emotional component, termed affective computing, has also received increased attention in recent years. Picard [1995] emphasizes the fact that "emotions play a necessary role not only in human creativity and intelligence, but also in rational human thinking and decision-making. Computers that will interact naturally and intelligently with humans need the ability to at least recognize and express affect."

From a theoretical standpoint, it seems reasonable to incorporate emotional awareness into systems designed to mimic (or produce) human-like creativity and intelligence, since emotions are such a basic part of being human. On a more practical level, affective displays on the part of a computerized agent can improve function and usability. Research has shown that incorporating emotional expression into the design of interactive agents can improve user engagement, satisfaction, and task performance [Klein et al., 2002, Partala and Surakka, 2004]. Users may also regard an agent more positively [Ochs et al., 2008] and consider it to be more believable [Bates, 1994] when it demonstrates appropriate emotional awareness.

A number of researchers have added an element of emotional awareness to automatic music arrangement and composition. For example, Delgado, Fajardo, and Molina-Solana [2009] use a rule-based system to generate compositions according to a specified mood. Rutherford and Wiggins [2003] analyze the features that contribute to the emotion of fear in a musical selection and present a system that allows for an input parameter that determines the level of "scariness" in the piece. Oliveira and Cardoso [2007] describe a wide array of features that contribute to emotional content in music and present a system that uses this information to select and transform chunks of music in accordance with a target emotion. Like these previously mentioned systems, our system is concerned with producing music with a desired emotional content, but rather than developing rule sets for different emotions, it uses statistical information in training corpora in order to generate emotion-specific accompaniments.

5.3 Methodology

In order to provide targeted accompaniment for a given story, each sentence in the text is first labeled with an emotion. For these experiments, selections are assigned labels of love, joy, surprise, anger, sadness, and fear, according the categories of emotions described by Parrott [2001]. Selections can also be labeled as neutral if the system finds no emotions present. Music is then generated to match the labels assigned by the system. The following sections briefly describe the labeling, music composition, and audio file generation processes.

5.3.1 Emotional Labeling of Text

A more detailed account of the emotional labeling system can be found in [Francisco and Hervás, 2007], but a brief description is provided here.

The system relies on a dictionary of word-to-emotion assignments obtained from a corpus of human-evaluated texts. Each of the texts in the corpus is marked by several people. The system then selects the emotion most often assigned to each sentence and uses this information to develop a data base of words (List of Emotional Words or LEW). For each of the relevant words in a given sentence (words whose part-of-speech is not included in a POS tag stop list) the system obtains its stem and associated emotion. The complement of the emotional content of words is also calculated. This information is then used to update probabilities in the LEW list of a particular word expressing a given emotion.

To account for words not occurring in the training corpus, the system relies on two additional resources: the ANEW word list and WordNet. ANEW, or the *Affective Norms for English Words* is a set of normative emotional ratings for a large number of words in the English language [Bradley and Lang, 1999]. Words have been rated in terms of pleasure, arousal, and dominance. WordNet is a semantic lexicon that groups English words into sets of synonyms called synsets [Fellbaum, 1998]. It can provide information about synonyms, antonyms, and hypernyms of a given word.

In order to tag a given piece of text, the system first performs sentence detection and tokenization to obtain the stem and the part-of-speech of each word and any words affected by negations. The system then determines the emotional value associated to each word or word negation by looking it up in the affective dictionary (LEW list) and using information from one of the additional resources if necessary. Once all the words of the sentences have been evaluated, the probability of conveying each emotion for the different words are summed

⟨*fear*⟩ When the lion came home that night, he stepped into the trap. ⟨/*fear*⟩
⟨*anger*⟩ He roared! ⟨/*anger*⟩
⟨*sadness*⟩ He wept! ⟨/*sadness*⟩
⟨*sadness*⟩ But he couldn't pull himself free. ⟨/*sadness*⟩
⟨*love*⟩ The mouse heard the lion's pitiful roar and came back to help him. ⟨/*love*⟩
⟨*neutral*⟩ The mouse eyed the trap and noticed the one thick rope that held it together. ⟨/*neutral*⟩
⟨*joy*⟩ She began nibbling and nibbling until the rope broke. ⟨/*joy*⟩
⟨*neutral*⟩ The lion was able to shake off the other ropes that held him tight. ⟨/*neutral*⟩
⟨*joy*⟩ He stood up free again! ⟨/*joy*⟩
⟨*neutral*⟩ The lion turned to the mouse and said, ⟨/*neutral*⟩
⟨*sadness*⟩ ”Dear friend, I was foolish to ridicule you for being small. ⟨/*sadness*⟩
⟨*joy*⟩ You helped me by saving my life after all!” ⟨/*joy*⟩

Figure 5.1: Example of emotional labels assigned to a portion of the story “The Lion and the Mouse”

and the emotion which has the highest probability is assigned to the sentence. Figure 5.1 shows the emotional labels assigned to a sample of text by the emotional labeling system.

5.3.2 Emotionally Targeted Music Generation

Further details on the process of generating music with targeted emotional content can be found in [Monteith et al., 2010], but a brief description is provided here.

In order to generate emotionally targeted music, corpora of songs representing the various emotions are constructed based on human evaluation of emotional content. Melodies are analyzed and *n*-gram models are generated representing what notes are most likely to follow a given series of notes in a given corpus. Statistics describing the probability of a melody note given a chord, and the probability of a chord given the previous chord, are also collected for each corpus.

The rhythm for the selection with a desired emotional content is generated by selecting a phrase from a randomly chosen selection in the corresponding data set and selecting and modifying a random number of measures. Once the rhythm is determined, pitches are selected for the melodic line according to the n-gram model constructed from the melody lines of the corpus with the desired emotion. At both the rhythm selection and melody generation steps in the process, the program generates a number of different possible candidates for melodies and rhythms, and neural network evaluators are used to evaluate the candidate phrases for similarity to human generated selections and selections with the target emotion.

Underlying harmony is determined using a Hidden Markov Model, with pitches considered as observed events and the chord progression as the underlying state sequence. The statistics for these probability distributions are gathered from the corpus of music representing the desired emotion. The accompaniment patterns for each of the selections in the various corpora are categorized, and the accompaniment pattern for a generated selection is probabilistically selected from the patterns of the target corpus. Common accompaniment patterns included arpeggios, chords sounding on repeated rhythmic patterns, and a low base note followed by chords on non-downbeats. Instruments for the melody and harmonic accompaniment are also probabilistically selected based on the frequency of various melody and harmony instruments in the corpus.

5.3.3 File Generation

Generating the actual audio files of the fairy tales with accompanying soundtrack was done following Algorithm 1. A text corpus is initially segmented at the sentence level (line 1) and each sentence is tagged with an emotion (line 2). Ten musical selections are generated for each possible emotional label and converted from MIDI to WAV format (lines 5-7) using WinAmp[1]. In order to produce a spoken version of a given fairy tale, each sentence is converted to an audio file (line 10) using FreeTTS,[2] an open-source text to speech program. This provides a collection from which musical accompaniments can be selected. Each audio phrase is analyzed to determine its length, and the musical file with matching emotional label that is closest in length to the sentence file is selected as accompaniment (lines 11-12).

[1] http://www.winamp.com

[2] http://freetts.sourceforge.net

Algorithm 1 *Algorithm for automatically generating soundtracks for text.* F is the text corpus (e.g. a fairy tale) for which a soundtrack is to be generated.

```
SoundTrack($F$)
 1: Divide $F$ into sentences: $S_1$ to $S_m$
 2: Assign emotion labels: $L_1$ to $L_m$
 3: $S' \leftarrow$ join consecutive sentences in $S$ with matching labels
 4: $L' \leftarrow$ join consecutive matching labels in $L$
 5: for all $L'_i$ in $L'$ do
 6:     Generate MIDI selections: $M_{i1}$ to $M_{i10}$
 7:     Convert to WAV files: $W_{i1}$ to $W_{i10}$
 8: end for
 9: for all $S'_i$ in $S'$ do
10:     $A_i \leftarrow$ Generate TTS audio recording from $S'_i$
11:     $k \leftarrow argmin_j |len(A_i) - len(W_{ij})|$
12:     $C_i \leftarrow A_i$ layered over $W_{ik}$
13: end for
14: $O \leftarrow C_1 + C_2 + ... + C_n$
15: return $O$
```

If all of the generated selections are longer than the audio file, the shortest selection is cut to match the length of the audio file. Since this is often the case, consecutive sentences with the same emotional label are joined before music is assigned (lines 3-4). Sentences labeled as "neutral" are left with no musical accompaniment. Finally, all the sentence audio files and their corresponding targeted accompaniments are concatenated to form a complete audio story (line 14).

5.4 Results

Musical accompaniments were generated for each of the following stories:[3]

- The Lion and the Mouse
- The Ox and the Frog
- The Princess and the Pea
- The Tortoise and the Hare
- The Wolf and the Goat

[3] All audio files used in these experiments are available at axon.cs.byu.edu/emotiveMusicGeneration

For comparison purposes, text-to-speech audio files were generated from the text of each story and left without musical accompaniment. (i.e. line 12 of Algorithm 1 becomes simply, $C_i \leftarrow A_i$.) Files were also generated in which each sentence was accompanied by music from a randomly selected emotional category, including the possibility of no emotion being selected (i.e. line 11 of Algorithm 1 becomes $k = rand(|L'| + 1)$, and file W_{i0} was silence for all i. Randomization was set such that $k = 0$ for approximately one out of three sentences.)

Twenty-four subjects were asked to listen to a version of each of the five stories. Subjects were divided into three groups, and versions of the stories were distributed such that each group listened to some stories with no music, some with randomly assigned music, and some with emotionally targeted music. Each version of a given story was played for eight people.

After each story, subjects were asked to respond to the questions listed in Figure 5.2. A Cronbach's alpha coefficient [Cronbach, 1951] was calculated on the responses of subjects in each group to test for inter-rater reliability. Coefficients for the three groups were $\alpha = 0.93$, $\alpha = 0.87$, and $\alpha = 0.83$. (Values over 0.80 are generally considered indicative of a reasonable level of reliability and consequently, a sufficient number of subjects for testing purposes.)

Table 5.1 shows the average ratings for selections in each of the three categories in response to the question "How much did you enjoy listening to the story?" On average, targeted music made a selection significantly more enjoyable than a version with random music. A Student's t-test reveals the significance level to be $p = 0.011$ for the difference in these two means. Selections in the "Targeted Music" group were also rated more enjoyable, on average, than selections in the "No Music" group, but the difference in means was not significant. Listeners did rate the version of "The Tortoise and the Hare" with emotionally targeted music as significantly more enjoyable than the "No Music" version ($p = 0.001$).

Table 5.2 reports the average ratings in response to the question "How effectively did the music match the events of the story?" Not surprisingly, music with targeted emotional

Please answer the following questions about the selection (1=low, 5=high):

- How much did you enjoy listening to the story?
- If music was included, how effectively did the music match the events of the story

Rate the intensity of any emotions present in the story (1=emotion was not present, 5=emotion was very present):

- Love
- Joy
- Surprise
- Anger
- Sadness
- Fear

Figure 5.2: Questionnaire used in experiments

content was rated significantly higher in terms of matching the events of the story than randomly generated music ($p = 0.003$).

Table 5.3 provides the intensity ratings for each of the six emotions considered, averaged over all five stories. Listeners tended to assign higher emotional ratings to selections in the "Random Music" category than they did to selections in the "No Music" category; however, this was not statistically significant. Average emotional ratings for the selections in the "Targeted Music" category had significantly higher ratings ($p = 0.027$) than selections accompanied by randomly generated music. When directly comparing "Targeted Music" with "No Music", average emotional ratings are again higher for the targeted music, though the difference falls a bit short of statistical significance at the $p = 0.05$ level ($p = 0.129$).

The difference in emotional ratings is even more pronounced when only emotions appearing as labels in the stories are considered. Table 5.4 shows the intensity ratings for "Love." This label only appears with high frequency in "The Wolf and the Goat." It

Table 5.1: Average responses to the question "How much did you enjoy listening to the story?"

	No Music	Random Music	Targeted Music
The Lion and the Mouse	2.88	2.13	2.75
The Ox and the Frog	3.50	2.75	3.00
The Princess and the Pea	3.00	3.38	4.13
The Tortoise and the Hare	2.75	2.75	3.88
The Wolf and the Goat	3.25	2.88	3.38
Average	3.08	2.78	3.43

Table 5.2: Average responses to the question "How effectively did the music match the events of the story?"

	Random Music	Targeted Music
The Lion and the Mouse	2.88	3.38
The Ox and the Frog	2.13	3.25
The Princess and the Pea	2.50	3.88
The Tortoise and the Hare	2.38	3.50
The Wolf and the Goat	1.75	3.25
Average	2.33	3.45

appears in two out of 31 sentences in "The Lion and the Mouse" and does not occur in any of the other stories. "The Wolf and the Goat" is the only story for which targeted emotional accompaniment resulted in higher intensity ratings for "Love" than did random or no accompaniment. Compare this to Table 5.5, which provides intensity ratings for "Joy," a label which appears in all stories, in most cases with fairly high frequency. For this emotion, targeted accompaniment was much more likely to result in increased intensity ratings.

Table 5.6 gives average intensity ratings when only labeled emotions are considered (compare to Table 5.3). In this analysis, selections in the "Targeted Music" category received higher intensity ratings than selections in both the "No Music" and "Random Music" categories, with both differences being very near statistical significance at the $p = 0.05$ level ($p = 0.056$ and $p = 0.066$, respectively). Note that the only emotional category in which targeted music does not tie or exceed the other two accompaniment styles in terms of intensity

Table 5.3: Average intensity of a given emotion for all stories

	No Music	Random Music	Targeted Music
Love	1.83	1.40	1.55
Joy	2.03	2.10	2.53
Surprise	2.63	2.50	2.75
Anger	1.48	1.60	1.55
Sadness	1.60	1.70	2.05
Fear	1.58	2.00	2.15
Average	1.85	1.88	2.10

Table 5.4: Average intensity ratings for "Love"

	No Music	Random Music	Targeted Music
The Lion and the Mouse	2.50	1.75	1.75
The Ox and the Frog	1.38	1.00	1.25
The Princess and the Pea	3.25	2.13	2.25
The Tortoise and the Hare	1.00	1.13	1.00
The Wolf and the Goat	1.00	1.00	1.50
Average	1.83	1.40	1.55

ratings is that of "Surprise." The fact that "Random Music" selections were rated as more surprising than "Targeted Music" selections is not entirely unexpected.

5.5 Discussion and Future Work

The component systems described here are arguably creative in their own right. Rabinow cites three components necessary for individual creativity: access to a "tremendous amount of information," an ability to "pull the ideas" or generate a lot of original work from that information, and the ability to "get rid of the trash." These elements are evident in the individual systems of this work. The text labeler is able to synthesize a large amount of information and use it to develop an awareness of human emotion. The music generation system similarly uses large quantities of data to direct its creative endeavors, generating a substantial number of melodies and evaluating them for desirability. But regardless of how creatively systems may behave on their own, Csikszentmihalyi argues that individual

Table 5.5: Average intensity ratings for "Joy"

	No Music	Random Music	Targeted Music
The Lion and the Mouse	2.875	1.875	3.125
The Ox and the Frog	1.375	2.125	1.5
The Princess and the Pea	2.875	2.875	3.375
The Tortoise and the Hare	2	2.25	2.75
The Wolf and the Goat	1	1.375	1.875
Average	2.03	2.10	2.53

Table 5.6: Average intensity of labeled emotions for all stories

	No Music	Random Music	Targeted Music
Love	1.75	1.38	1.75
Joy	2.03	2.10	2.53
Surprise	2.67	2.88	2.75
Anger	1.56	1.50	1.56
Sadness	1.94	2.06	2.31
Fear	1.94	2.13	2.31
Average	1.98	2.01	2.20

actions are insufficient to assign the label of "creative" in and of themselves. As he explains, "...creativity must, in the last analysis, be seen not as something happening within a person but in the relationships within a system." In other words, an individual has to interact with and have an impact on a community in order to be considered truly creative.

Csikszentmihalyi continues with an example from history: "According to the systems model, it makes perfect sense to say that Raphael was creative in the sixteenth and in the nineteenth centuries but not in between or afterward. Raphael is creative when the community is moved by his work, and discovers new possibilities in his paintings." While it is difficult for computer programs to be "moved" by other programs per se, the combination of the systems discussed in this paper does allow for a broader array of possibilities for computerized creative works. Adding the ability to label emotions in text allows for generated music to be targeted to a specific project rather than simply existing in a vacuum. The application described in this paper is one example of how music generated by the system can be put to practical use through the assistance of emotion-labeled text. Other possibilities include writing accompanying music for lyrics or soundtracks for films with a provided script. The system could also be used in conjunction with other "creative" computational systems. For example, it could be used to generate targeted emotional soundtracks for computer-composed stories or automatically generated computer games.

In addition to allowing further interaction with the "society" of creative programs, our combination of systems also allows creative works to have a greater impact on humans. Music can have a significant effect on human perception of a story. However, as demonstrated in previous literature and in the results of our study, this impact is most pronounced when music is well-matched to story content. Music generated without regard to the emotional content of the story appears to be less effective both at eliciting emotion and at making a story more enjoyable for listeners.

Future work on this project will involve improving the quality of the generated audio files. Some of the files generated with the text-to-speech program were difficult to understand. The lack of clarity may have had an impact on the overall results of the survey, since adding music to the files could add to the problem of unintelligibility. Perhaps some listeners were more likely to prefer the versions of stories without soundtracks simply because they were easier to understand. A clearer reading, either by a different text-to-speech program or a recording of a human narrator, would likely enhance the intelligibility and possibly result in higher enjoyability ratings for the accompanied stories. Future work will also include adding more sophisticated transitions between musical selections in the accompaniment. This may also improve the quality of the final audio files.

Chapter 6

Automatic Generation of Melodic Accompaniments for Lyrics

> ***"If a composer could say what he had to say in words he would not bother trying to say it in music."*** **–Gustav Mahler**

K. Monteith, T. Martinez, and D. Ventura. Automatic Generation of Melodic Accompaniments for Lyrics. In *Proceedings of the Third International Conference on Computational Creativity*, pages 87-94, 2012.

Abstract: *Music and speech are two realms predominately species-specific to humans, and many human creative endeavors involve these two modalities. The pairing of music and spoken text can heighten the emotional and cognitive impact of both - the complete song being much more compelling than either the lyrics or the accompaniment alone. This work describes a system that is able to automatically generate and evaluate musical accompaniments for a given set of lyrics. It derives the rhythm for the melodic accompaniment from the cadence of the text. Pitches are generated through the use of n-gram models constructed from melodies of songs with a similar style. This system is able to generate pleasing melodies that fit well with the text of the lyrics, often doing so at a level similar to that of human ability.*

6.1 Introduction

Programmers and researchers have often attempted to endow machines with some form of intelligence. In some cases, the end goal of this is purely practical; a machine with the capacity to learn could provide a multitude of useful and resource-saving tasks. But in other cases, the goal is simply to make machines behave in a more creative or more "human" manner. As one author explains, "Looked at in one way, ours is a history of self-imitation...We are ten times more fascinated by clockwork imitations than by real human beings performing the same task" McCorduck [2004].

One major area of human creativity involves the production of music. Wiggins [2006] states that, "...musical behavior is a uniquely human trait...further, it is also ubiquitously human: there is no known human society which does not exhibit musical behaviour in some form." Naturally, many computer science researchers have turned their attention to musical computation tasks. Researchers have attempted to classify music, measure musical similarity, and predict the musical preferences of users [Chai and Vercoe, 2001, McKay and Fujinaga, 2004]. Others have investigated the ability to search through, annotate, and identify audio files [Dannenberg et al., 2003, Dickerson and Ventura, 2009]. More directly in the realm of computational creativity, researchers have developed systems that can automatically arrange and compose music [Oliveira and Cardoso, 2007, Delgado et al., 2009].

Like music, speech is an ability that is almost exclusively human. While species such as whales or birds may communicate through audio expressions, and apes may even be taught simple human-like vocabularies and grammars using sign language, the complexities of human language set us apart in the animal kingdom. Major research efforts have been directed toward machine recognition and synthesis of human speech [Koskenniemi, 1984, Rabiner, 1989] Computer programs have been designed to carry on conversations, some of them doing so in a surprisingly human-like manner [Weizenbaum, 1966, Saygin et al., 2000]. More creative programming endeavors have involved the generation of poetry [Gervás, 2001, Rahman and Manurung, 2011] or text for stories [Riedl, 2004, Pérez y Pérez and Sharples, 2004, Gervás et al., 2005, Ang et al., 2011].

Gfeller [1990] points out the similarities between speech and music: "Both speech and music are species specific and can be found in all known cultures. Both forms of communication evolve over time and have structural similarities such as pitch, duration, timbre, and intensity organized through particular rules (i.e. syntax or grammar) that result in listener expectations." Studies show that music and the spoken word can be particularly powerful when paired together. For example, in one study, researchers found that a sung version of a story was often more effective at reducing an undesirable target behavior than a read version of the story [Brownell, 2002]. Music can help individuals with autism and auditory processing disorders more easily engage in dialog [Wigram, 2002]. The pairing of music with language can even help individuals regain lost speech abilities through a process known as Melodic Intonation Therapy [Gfeller, 1990, Schlaug et al., 2008]. On the other hand, lyrics have the advantage of being able to impart discursive information where the more abstract nature of music makes it less fit to do so [Kreitler and Kreitler, 1972]. Lyrics can also contribute to the emotional impact of a song. One study found that lyrics enhanced the emotional impact of a selection with sad or angry music [Ali and Peynircioglu, 2006]. Another found that lyrics tended to be a better estimator of the overall mood of a song than the melody when the lyrics and the melody disagree [Wu et al., 2009].

This work describes a system that can automatically compose melodic accompaniments for any given text. For each given lyric, it generates hundreds of different possibilities for rhythms and pitches and evaluates these possibilities with a number of different metrics in order to select a final output. The system also incorporates an awareness of musical style. It learns stylistic elements from a training corpus of melodies in a given genre and uses these to output a new piece with similar elements. In addition to self-evaluation, the generated selections are further evaluated by a human audience. Survey feedback indicates that the system is able to generate melodies that fit well with the cadence of the text and that are often as pleasing as the original accompanying tunes. Colton, Charnley, and Pease [2011] suggest a number of different measures that can be used to evaluate systems during the creative process. We direct particular attention to two of these–precision and reliability–and demonstrate that, for simpler styles, our system is able to perform well with regard to these metrics.

6.2 Related Work

Conklin [2003] summarizes a number of statistical models which can be used for music generation, including random walk, Hidden Markov Models, stochastic sampling, and pattern-based sampling. These approaches can be seen in a number of different studies. For example, Chuan and Chew [2007] use Markov chains to harmonize given melody lines, focusing on harmonization in a given style. Cope [2006] also uses statistical models to generate music in a particular style, producing pieces indistinguishable from human-generated compositions. Pearce and Wiggins [2007] provide an analysis of a number of strategies for melodic generation, including one similar to the generative model used in this paper.

Delgado, Fajardo, and Molina-Solana [2009] use a rule-based system to generate compositions according to a specified mood. Oliveira and Cardoso [2007] describe a wide array of features that contribute to emotional content in music and present a system that uses this information to select and transform chunks of music in accordance with a target emotion.

Researchers have also directed efforts towards developing systems intended for accompaniment purposes. Dannenberg [1985] presents a system of automatic accompaniment designed to adapt to a live soloist. Lewis [2000] also details a "virtual improvising orchestra" that responds to a performer's musical choices.

While not directly related to generating melodic accompaniment for lyrics, a number of studies have looked at aligning musical signals to textual lyrics (the end result being similar to manually-aligned karaoke tracks). For example, Wang and associates [2004] use both low-level audio features and high-level musical knowledge to find the rhythm of the audio track and use this information to align the music with the corresponding lyrics.

6.3 Methodology

In order to generate original melodies, a set of melodies is compiled for each different style of composition. These melodies were isolated from MIDIs obtained from the Free MIDI

File Database[1] and the "I Love MIDIs" website[2]. These selections help determine both the rhythmic values and pitches that will be assigned to each syllable of the text. The system catalogs the rhythmic patterns that occur for each of the various numbers of notes in a given measure. The system also creates an n-gram model representing what notes are most likely to follow a given series of notes in a given set of melodies. Models were developed for three stylistic categories: nursery rhymes, folk songs (bluegrass), and rock songs (Beatles).

For each lyric, the system first analyzes the text and assigns rhythms. It determines where the downbeats will fall for each given line of the text. One hundred different downbeat assignments are generated randomly, and evaluated according to a number of aesthetic measures. The system selects the random assignment with the highest score for use in the generated melody. The system then determines the rhythmic values that will be assigned to each syllable in the text by counting the number of syllables in a given measure and finding a rhythm that matches that number of syllables in one of the songs of the training corpus. Once rhythmic values are assigned, the system assigns pitches to each value using the n-gram model constructed from the training corpus. Once again, one hundred different assignments are generated and evaluated according to a number of metrics. Further details on the rhythm and pitch generation are provided in the following subsections.

6.3.1 Rhythm Generation

Rhythms are generated based on patterns of syllabic stress in the lyrics. Each word of the text is located in the CMU Pronunciation Dictionary[3] to determine the stress patterns of the constituent phonemes. (Each phoneme in the dictionary is labeled 0, 1, or 2 for "No Stress," "Primary Stress," or "Secondary Stress.") The system also looks up each word to determine if it occurs on a list of common articles, prepositions, and conjunctions.

[1] http://www.mididb.com/

[2] http://www.ilovemidis.com/ForKids/NurseryRhymes/

[3] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

Lyrics:	Pat	a	cake	pat	a	cake	ba-	ker's	man
Phonemes:	PAET	AH	KEYK	PAET	AH	KEYK	BEY	KERZ	MAEN
Stress:	1	0	1	1	0	1	1	0	1
Downbeats:	true	false	false	true	false	false	true	false	true

Figure 6.1: Sample downbeat assignments for *Pat-A-Cake* lyrics

The system then attempts to find the best positions for downbeats. For each given line of text, the system generates 100 possible downbeat assignments. The text of each line is distributed over four measures, so four syllables are randomly selected to carry a downbeat. Each assignment is then scored, and the system selects the assignment receiving the highest score for use in the melodic accompaniment. Downbeat assignments that fall on stressed syllables are rated highly, as are downbeats that fall on the beginning of a word and ones that do not fall on articles, prepositions, or conjunctions. Downbeat assignments that space syllables more evenly across the allotted four measures are also rated more highly (i.e. assignments that have a lower standard deviation for number of syllables per measure receive higher scores). See Figure 6.5 for further details on the precise downbeat scoring metrics. Figure 6.1 illustrates a possible downbeat assignment for a sample lyric.

Once the downbeats are assigned, a rhythmic value is assigned to each syllable. The system randomly selects a piece in the training corpus to provide rhythmic inspiration. This selection determines the time signature of the generated piece (e.g. three beats or four beats to a measure). For each measure of music generated, the system looks to the selected piece and randomly chooses a measure that has the necessary number of notes. For example, if the system needs to generate a rhythm for a measure with three syllables, it randomly chooses a measure in the training corpus piece that has three notes in it and uses its rhythm in the generated piece. If no measures are available that match the number of syllables in the lyric, the system arbitrarily assigns rhythmic values, with longer values being assigned to earlier syllables. For example, in a measure with three syllables using a three-beat pattern, each syllable would be assigned to a quarter note. In a measure with four syllables, the first

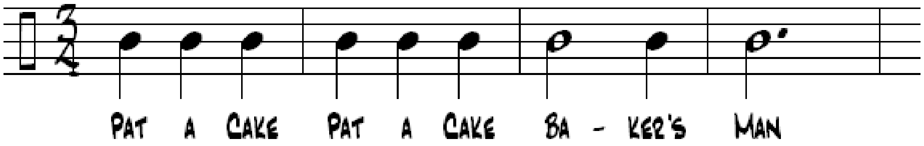


Figure 6.2: Default rhythm assignments for *Pat-A-Cake* lyrics

two syllables would be assigned to quarter notes and the last two syllables to eighth notes. Figure 6.2 illustrates the default rhythms assignment for a sample lyric.

6.3.2 Pitch Generation

Once the rhythm is determined, pitches are selected for the various rhythmic durations. Selections from a given style corpus are first transposed into the same key. Then an n-gram model with an n value of four is constructed from these original melodic lines. The model was created simply from the original training melodies, with no smoothing. For the new, computer-generated selections, melodies are initialized with a series of random notes, selected from a distribution that models which notes are most likely to begin musical selections in the given corpus. In order to foster song cohesion, each line of the song is initialized with the same randomly generated three notes. Additional notes for each line are randomly selected based on a probability distribution of what note is most likely to follow the given three notes as indicated by the n-gram model of the style corpus.

The system generates several hundred possible series of pitches for each line. Each possible pitch assignment is then scored. To encourage melodic interest, higher scores are given to melodic lines with a higher number of distinct pitches and melodies featuring excessive repeated notes are penalized. Melodies with a range greater than an octave and a half or with interval jumps greater than an octave are penalized since these are less “sing-able.” Melodic lines that do not end on a note in a typical major or minor scale and final melodic lines that do not end on a tonic note are given a score of zero. More precise details about

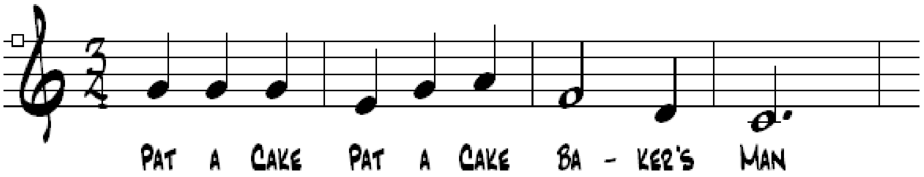


Figure 6.3: Sample pitch assignments for *Pat-A-Cake* lyrics

the scoring of pitch assignments are given in Figure 6.5. Possible pitch assignments for a sample lyric are shown in Figure 6.3.

6.4 Results

Accompaniments were generated for lyrics in three stylistic categories: nursery rhymes, folk songs (bluegrass), and rock songs (Beatles). In each case, an attempt was made to find less commonly known melodies, so that the generated music could be more fairly compared to the original melodic lines. Melodic lines were generated for the following:

Nursery rhymes:

- Goosey Goosey Gander
- Little Bo Peep
- Pat-a-Cake
- Rub-a-Dub-Dub
- The Three Little Kittens

Folk songs:

- Arkansas Traveler
- Battle of New Orleans
- Old Joe Clark
- Sally Goodin
- Wabash Cannonball

Rock songs:

- Act Naturally
- Ask Me Why

- A Taste of Honey
- Don't Pass Me By
- I'll Cry Instead

Three melodies were generated for each of the fifteen lyrics considered. One was generated using a corpus of songs that matched the style of the lyrics (e.g. to generate a melody for *Goosey Goosey Gander* the four other nursery rhyme songs were used to build the n-gram model) and two more were generated in the remaining two creative styles[4].

Study participants were divided into four groups. Each group was asked to listen to versions of songs for each of the fifteen lyrics, with selections for each group being a mixture of lyrics with the original human-composed melodies and lyrics with the three types of computer-generated melodies. Subjects were not informed that any of the melodies were computer-generated until after data collection. Fifty-two subjects participated in the study, and each melodic version was played for thirteen people.

After each selection, subjects were asked to respond to the following questions (1=not at all, 5=very much):

- How familiar are you with these lyrics? 1 2 3 4 5
- How familiar are you with this melody? 1 2 3 4 5
- How pleasing is the melodic line? 1 2 3 4 5
- How well does the music fit with the lyrics? 1 2 3 4 5
- Is this the style of melody you would have expected to accompany these lyrics? 12345
- Are you familiar with any other melodies for these lyrics? YES NO

Table 6.1 shows the average responses to the question about familiarity of lyrics for each of the three categories. In each case, lyrics were rated as more familiar when they were paired with their original melodies as opposed to the computer-generated melodies. However, none of these differences were significant at the $p < 0.05$ level. The majority of subjects were relatively unfamiliar with the bluegrass and rock lyrics. The nursery rhyme lyrics were slightly more familiar, but in many cases, subjects were familiar with the lyrics but not any specific tune.

[4] Selections generated for these experiments are available at http://axon.cs.byu.edu/emotiveMusicGeneration

Table 6.1: Average responses to the question "How familiar are you with these lyrics?" Each row represents a compositional style and each column a category of lyrics.

	bluegrass	nursery	rock	average
bluegrass	1.34	3.09	1.19	1.87
nursery	1.14	3.32	1.19	1.88
rock	1.25	3.28	1.11	1.88
original	1.50	3.50	1.47	2.16

Table 6.2: Average responses to the question "How familiar are you with this melody?" Each row represents a compositional style and each column a category of lyrics.

	bluegrass	nursery	rock	average
bluegrass	1.62	1.49	1.40	1.50
nursery	1.53	2.17	1.34	1.68
rock	1.41	1.39	1.24	1.35
original	2.31	2.94	1.81	2.35

Table 6.2 shows the average responses to the question about familiarity of melody for each of the three categories. On average, subjects were slightly more familiar with the original melodies in the bluegrass and rock categories than they were with the lyrics. The original nursery rhymes melodies were rated as slightly less familiar on average than the lyrics. System-generated melodies received an average score of less than two for familiarity in each of the three categories (significantly lower than original melodies with a statistical significance of $p < 0.01$).

Subjects were likely to be less receptive to new melodies if they were very familiar with the old ones. (One respondent mentioned that hearing a new melody to a familiar childhood song was a little "unnerving".) Tables 6.3 through 6.7 report only the responses where subjects indicated that they were not familiar with an alternate melody for a given set of lyrics.

As shown in Table 6.3, the system was able to generate melodies that received the same average ratings for pleasing melodic lines as the original melodies. The average rating for songs in the bluegrass style was almost identical to that of the original melodies. The average ratings for pleasantness of generated nursery rhythm melodies was not significantly different than the original tunes.

Table 6.3: Average responses to the question "How pleasing is the melodic line?" Each row represents a compositional style and each column a category of lyrics.

	bluegrass	nursery	rock	average
bluegrass	3.50	3.50	3.56	3.52
nursery	3.37	3.24	3.09	3.23
rock	2.70	2.17	2.16	2.34
original	3.79	3.79	2.95	3.51

Table 6.4: Average responses to the question "How pleasing is the melodic line?" for six songs where system-generated melody in one or more styles scored higher than the original melody.

	Battle of New Orleans	Little Bo Peep	Rub A Dub Dub	Act Naturally	Ask Me Why	I'll Cry Instead
bluegrass	3.23	**3.60**	**3.80**	**3.50**	**4.23**	**3.79**
nursery	**3.92**	**3.43**	3.17	2.91	**3.14**	**2.92**
rock	2.83	2.60	2.13	2.54	2.00	**2.36**
original	3.33	3.22	3.50	2.70	2.83	2.12

For over a third of the lyrics, a computer-generated melody in at least one style was rated as more pleasing than the original melody. These tunes are listed in Table 6.4 along with their average ratings. For example, the original melody for *Battle of New Orleans* received a rating of 3.33 for average melodic pleasantness. The computer-generated melody for this lyric in a nursery rhyme style received a rating of 3.92. The original melody for *Little Bo Peep* received an average melodic pleasantness rating of 3.22. The bluegrass-styled computer-generated melody received a rating of 3.80, and the nursery-rhyme-styled generated melody received a rating of 3.43.

Table 6.5 shows that the original melodies were rated on average as fitting a little better with the lyrics (although the difference between the original melodies and the songs composed in the bluegrass style is not statistically significant). However, as shown in Ta-

Table 6.5: Average responses to the question "How well does the music fit with the lyrics?" Each row represents a compositional style and each column a category of lyrics.

	bluegrass	nursery	rock	average
bluegrass	3.59	3.20	3.18	3.32
nursery	3.35	3.36	2.71	3.14
rock	3.23	2.18	2.26	2.56
original	3.88	4.27	2.90	3.68

Table 6.6: Average responses to the question "How well does the music fit with the lyrics?" for six songs where system-generated melody in one or more styles scored higher than the original melody.

	Arkansas Traveler	Old Joe Clark	Three Little Kittens	Ask Me Why	A Taste of Honey	I'll Cry Instead
bluegrass	**4.08**	2.71	**4.25**	**3.54**	2.57	**3.43**
nursery	3.08	2.75	3.80	**3.07**	**2.85**	**2.38**
rock	3.08	**3.00**	2.18	1.77	2.08	**2.27**
original	3.91	2.75	4.17	2.75	2.79	2.15

ble 6.6 a number of the individual computer-generated melodies were still rated as fitting better with the lyrics than the original melodies. For example, the rock version of *Old Joe Clark* received a rating of 3.00 from this metric while the original version received a rating of 2.75. Both the bluegrass and nursery-rhyme versions of *Ask Me Why* received higher ratings than the original version.

Table 6.7 reports responses to the question "Is this the style of melody you would have expected to accompany these lyrics?" Not surprisingly, the original melodies were more "expected" on average than melodies composed in new styles. The computer-generated melodies composed in the style of the original melodies were also generally more expected with one exception: bluegrass melodies for rock lyrics tended to receive higher expectation ratings.

In a number of cases, the system was able to compose an unexpected melody that still received high ratings for pleasing melodies and a lyric/note match. Two such examples

Table 6.7: Average responses to the question "Is this the style of melody you would have expected to accompany these lyrics?"

	bluegrass	nursery	rock	average
bluegrass	3.47	2.85	2.91	3.08
nursery	3.22	3.46	2.44	3.04
rock	3.12	1.82	2.14	2.36
original	3.69	4.27	2.79	3.58

Table 6.8: Average responses to questions for two songs where the melodic accompaniment was surprising but still worked.

	Pat-A-Cake (bluegrass)	Act Naturally (bluegrass)
How pleasing is the melodic line?	3.80	3.50
How well does the music fit with the lyrics?	3.20	3.17
Is this the style of melody you would have expected?	2.60	2.50

are shown in Table 6.8. In both cases, the songs received above average ratings for melodic pleasantness and average ratings for music/lyric match, but below average ratings for style expectedness.

6.5 Discussion

The original nursery rhymes were composed predominantly with notes of the major scale, and the rhythms in these songs were similarly simple. (Songs generated with corpus-inspired rhythms were quite similar to songs generated with the system's default rhythms.) With the exception of a flat seventh introduced by the mixolydian scale of *Old Joe Clark*, the bluegrass melodies also feature pitches exclusively from the major scale. Bluegrass rhythms also tended to be similarly straightforward. With simpler rhythms and fewer accidentals, more of the melodies generated in these two styles are likely to "work." The original bluegrass melodies tended to have more interesting melodic motion, and this appears to have translated into more interesting system-generated melodies. In contrast, the rock songs featured a much wider variety of scales and accidentals. These extra tones do add color to the generated selections, but further refinements may be necessary to select which more complicated melodies are "fresh" or "original" instead of just "weird."

Wiggins [2006] proposes a definition for computational creativity as "The performance of tasks which, if performed by a human, would be deemed creative." The task of simply composing any decent new melody for an established tune could be considered creative. Composing one that improved on the original constitutes an even greater degree of creative talent. By this metric, our system fits the definition of "creative."

Colton [2008] suggests that, for a computational system to be considered creative, it must be perceived as possessing skill, appreciation, and imagination. A basic knowledge of traditional music behavior allows a system to meet the "skillful" criteria. Our system takes advantage of statistical information about rhythms and melodic movement found in the training songs to compose new melodies that behave according to traditional musical conventions. A computational system may be considered "appreciative" if it can produce something of value and adjust its work according the preferences of itself or others. Our system addresses this criterion by producing hundreds of different possible rhythm and pitch assignments and evaluating them against some basic rules for pleasantness and singability. The "imaginative" criterion can be met if the system can create new material independent of both its creators and other composers. Since all of the generated melodies can be distinguished from songs in the training corpora, this criterion is met at least on a basic level. Our system further demonstrates its imaginative abilities by composing melodies in alternate styles that still manage to demonstrate an acceptable level of melodic pleasantness and synchronization with the cadence of the text.

Boden [1995] argues that unpredictability is also a critical element of creativity, and a number of researchers have investigated the role of unpredictability in creative systems [Macedo, 2001, Macedo and Cardoso, 2002] Our system meets the requirement of unpredictability with its ability to compose in various and sometimes unexpected styles. It is able to generate melodies that surprise listeners but still achieve high ratings for pleasantness.

Colton, Charnley, and Pease [2011] propose a number of different metrics in conjunction with their FACE and IDEA models that can be used to assess software during a session of creative acts. Equations for calculating these metrics are listed in Figure 6.4, were S is the creative system, (c_i^g, e_i^g) is a concept/expression pair generated by the system, $\overline{a^g}$ is an aesthetic measure of evaluation, and t is a minimum acceptable aesthetic threshold. Two of the measures suggested are precision (obtained by dividing the number of generated works by the number that met a minimum acceptable aesthetic level) and reliability (obtained from taking the system's best creation as calculated by some aesthetic measure and subtracting the system's worst). Table 6.9 reports the results of these calculations for the system's compositions in each of the three styles and compares them to the same metrics calculated for the original songs using responses to the question "How pleasing is the melodic line?" as the scoring metric. In order to calculate precision, we consider the worst score obtained by an original, human-composed melody to be the minimum acceptable threshold value. While the prize for most pleasing melody still goes to a human-composed song, all of the songs composed in a bluegrass and nursery style and two-thirds of the rock songs meet the basic criteria of being better than the worst original melody. The system is generating original melodies that are better than some established, human-generated songs a remarkable percentage of the time. The reliability of the system in generating bluegrass and nursery-style melodies is also worth mentioning. The reliability measures for these two categories are 1.30 and 1.33 as compared to the 2.38 reliability measure for original songs. (Note that, for reliability, smaller scores are more desirable.) While the system probably shouldn't quit its

$$\begin{gathered}
average(S) = \frac{1}{n}\sigma_{i=1}^{n}\overline{a^g}(c_i^g, e_i^g) \\
best\_ever(S) = max_{i=1}^{n}(\overline{a^g}(c_i^g, e_i^g)) \\
worst\_ever(S) = min_{i=1}^{n}(\overline{a^g}(c_i^g, e_i^g)) \\
precision(S) = \frac{1}{n}|\{(c_i^g, e_i^g) : 1 < i < n \wedge \overline{a^g}(c_i^g, e_i^g) > t\}| \\
reliability(S) = best\_ever(S) - worst\_ever(S)
\end{gathered}$$

Figure 6.4: Assessment metrics proposed by Colton, Charnley, and Pease [2011]

Table 6.9: Assessment metrics calculate on average responses to the question "How pleasing is the melodic line?"

	bluegrass	nursery	rock	original
average	3.52	3.23	2.34	3.51
best ever	4.23	3.92	3.83	4.50
worst ever	2.93	2.58	1.73	2.12
precision	1.00	1.00	0.67	1.00
reliability	1.30	1.33	2.11	2.38

day job to become a classic rock songwriter quite yet, it is considerably reliable at producing reasonable and pleasing melodies in the other two genres.

Similar results can be seen in Table 6.10 where responses to the question "How well does the music fit with the lyrics?" are used as the aesthetic measure. As with the previous calculations, the "worst ever" score for an original melody was used as a minimum aesthetic threshold for the generated melodies. Again, all of the nursery rhyme and bluegrass-styled compositions meet this threshold, as do two-thirds of the rock-styled songs. A song generated in the nursery rhyme or bluegrass style also more reliably matches the lyrics than an arbitrarily selected human-generated song.

Previous versions of our system analyzed each melody in a given training corpus according to a number of different metrics and used the results in the construction of neural networks designed to evaluate generated melodies [Monteith et al., 2010]. For the sake of simplicity and computational speed, the most pertinent of these findings were distilled into rules for use by the system in these experiments. In other words, the information gathered by the system to date about melody generation has been simplified and codified so that more focus could be directed towards matching rhythms to text. However, the system could likely benefit from the use of additional metrics and further "observation" of human-generated and

Table 6.10: Assessment metrics calculate on average responses to the question "How well does the music fit with the lyrics?"

	bluegrass	nursery	rock	original
average	3.32	3.14	2.56	3.68
best ever	4.25	3.86	4.23	4.75
worst ever	2.57	2.36	1.63	2.15
precision	1.00	1.00	0.67	1.00
reliability	1.68	1.49	2.61	2.60

approved tunes in its attempts to create pleasing melodies. A similar process of evaluation could be applied to the process of rhythm generation, particularly in the assignment of downbeats. Currently, the system relies on a small set of arbitrary, pre-coded rules to determine downbeat placement. It would likely require a much larger training corpus than we currently have available, but perhaps more natural-sounding placements could be obtained if the system could learn from a corpus of "good" lyric/melody pairings the types of words and syllables best suited for supporting downbeats. Audience feedback could help determine an optimal weighting of the various evaluation criteria.

```
MelodicAccompaniment($Lyric$,$StyleCorpus$)
for all $LINE_i$ in $Lyric$ do
    $STR_i \leftarrow$ patterns of syllabic stress in $LINE_i$
    $POS_i \leftarrow$ parts of speech for each syllable in $LINE_i$
    $BEG_i \leftarrow$ boolean values indicating that a syllable in $LINE_i$ begins a word
    for $i = 1 \rightarrow 100$ do
        $DB_j \leftarrow$ randomly assign downbeats to four syllables
        $score_j \leftarrow$ ScoreDownbeats($DB_j$,$STR_i$,$POS_i$,$BEG_i$)
    end for
    $DB_i \leftarrow DB_j$ that coincides with the largest $score_j$
    $RHYTHM_i \leftarrow$ SelectRhythms($DB_i$)
    for $i = 1 \rightarrow 100$ do
        $PITCHES_j \leftarrow$ assign pitches using $n$-gram model from $StyleCorpus$
        $score_j \leftarrow$ ScorePitches($PITCHES_j$)
    end for
    $PITCHES_i \leftarrow PITCHES_j$ that coincides with the largest $score_j$
    $MELODY_i \leftarrow$ combine $RHYTHM_i$ and $PITCHES_i$
    $MELODY+ = MELODY_i$
end for
return $MELODY$
```

```
ScoreDownbeats($DB_j$,$STR_i$,$POS_i$,$BEG_i$)
for $k = 1 \rightarrow j$ do
    If $DB_{jk}$ and $STR_{ik} = 1$ then $score+ = 1$
    If $DB_{jk}$ and $POS_{ik}! = Art|Prep|Conj$ then $score+ = 0.5$
    If $DB_{jk}$ and $BEG_{ik}$ then $score+ = 0.5$
    $x \leftarrow maxSyllablesPerMeasure$
    $score+ = (x - stdDevSyllablesPerMeasure) * 0.5$
    $score+ = (x - numPickupSyllables) * 0.25$
    $score+ = (x - numSyllablesLastMeasure) * 0.25$
end for
return score
```

```
SelectRhythms($D_i$, $S_i$)
$M \leftarrow$ divide $S_i$ into measures based on $D_i$
$C \leftarrow$ randomly select a song in $StyleCorpus$
$R \leftarrow 0$
for all $M_j$ in $M$ do
    $R_j \leftarrow$ randomly selected measure from $C$ with the same # of notes as syllables in $M_j$
    $R$ += $R_j$
end for
return $R$
```

```
ScorePitches($PITCHES_j$)
$score \leftarrow uniquePitches(PITCHES_j)/size(PITCHES_j)$
If $MaxRepeatPitches(PITCHES_j) < maxRepeatPitches$ then $score+ = 1$
If $Range(PITCHES_j) < maxRange$ then $score+ = 1$
If $MaxInterval(PITCHES_j) < maxInterval$ then $score+ = 1$
If $!EndsOnScaleNote(PITCHES_j)$ then $score = 0$
If $LastLine(j)$ and $!EndsOnTonic(PITCHES_j)$ then $score = 0$
return score
```

Figure 6.5: Algorithm for automatically generating melodic accompaniment for text

Part III

Additional Machine Learning Research

"No, I'm not interested in developing a powerful brain. All I'm after is just a mediocre brain, something like the President of the American Telephone and Telegraph Company." **–Alan Turing**

Chapter 7

Aggregate Certainty Estimators

"Democracy...is a charming form of government, full of variety and disorder; and dispensing a sort of equality to equals and unequals alike." **–Plato**

K. Monteith and T. Martinez. Aggregate Certainty Estimators. To appear in *Computational Intelligence.*

Abstract: *Selecting an effective method for combining the votes of base inducers in a multi-classifier system can have a significant impact on the system's overall classification accuracy. Some methods cannot even achieve as high a classification accuracy as the most accurate base classifier. To address this issue, we present the strategy of Aggregate Certainty Estimators, which uses multiple measures to estimate a classifier's certainty in its predictions on an instance-by-instance basis. Use of these certainty estimators for vote-weighting allows the system to achieve a higher overall average in classification accuracy compared to the most accurate base classifier. Weighting with these aggregate measures also results in higher average classification accuracy than weighting with single certainty estimates. Aggregate Certainty Estimators outperform three baseline strategies, as well as the methods of Modified Stacking and Arbitration, in terms of average accuracy over 36 data sets.*

7.1 Introduction

In the late eighteenth century, the Marquis de Condorcet composed the *Essay on the Application of Analysis to the Probability of Majority Decisions.* It outlined the concept now known as the "Condorcet Jury Theorem," the idea that, if each member of a group has a greater than 50% chance of making a correct decision, the probability that a plurality of voters will make the correct decision increases as voters are added. The essay was originally intended to provide a theoretical argument for the benefits of democracy. However, the concept also has application in the field of supervised learning. In theory, a group of classifiers should be better suited to the task of classification. The base classifiers need not even be highly accurate. As long as each classifier can exhibit a better-than-random performance, an ensemble of these classifiers should be able to take advantage of the expertise of each in order to assign more accurate classifications. Schapire [1990] demonstrated how a collection of "weak" classifiers—ones that perform only slightly better than random guessing—can be combined to produce a classifier with arbitrarily high accuracy. This provided the theoretical basis for his well-known AdaBoost algorithm, [Freund and Schapire, 1996, Schapire and Singer, 1998]. Other researchers have proposed similar ensemble-creation strategies, ranging from the simple strategy of Bagging [Breiman, 1996] to Random Forests [Breiman, 2001] and Bayesian Model Averaging [Hoeting et al., 1999].

Rokach [2010] outlines four basic building blocks of an ensemble: a labeled training set, a base inducer, a diversity generator, and a combiner. A number of strategies can be used in selecting the base inducers and generating diversity. For example, with the Adaptive Mixture of Local Experts strategy [Jacobs et al., 1991], a gating network determines the probability of selecting the output of one of the base inducers. Delegating [Ferri et al., 2004] is an approach where a base inducer assigns the final class label to a given instance only if it has high certainty in that particular class. If it is less confident, the instance is delegated to another base inducer. Diversity can be generated by selecting different subsets of the training set for use in training each of the base inducers. With Bagging [Breiman, 1996], instances are drawn randomly with replacement from the original training set to create a new set with which to train the base inducers. The AdaBoost algorithm [Freund and Schapire, 1996, Schapire and Singer, 1998] takes into account which instances were misclassified by previously constructed base inducers when selecting data for training subsequent ones. Multi-classifier systems address the problem of diversity generation by using different algorithms for training their base inducers. For example, Ho et al. [1994] use four different algorithmic approaches to character recognition and discuss methods of combining their outputs.

This work focuses on the fourth aspect outlined by Rokach: combining the outputs of the base inducers. It investigates the possibility that the same principle of combining weak classifiers to produce a strong one can also be applied to the weighting strategies used when combining the votes of those classifiers. While, as in the case of Naive Bayes, there is often a standard method for estimating confidence in a classifier's predictions, certainty in classification can be estimated in a variety of ways. If each of these methods can demonstrate even a slight tendency for assigning higher certainly values to correctly classified instances as opposed to incorrectly classified instances, they could theoretically be combined to create a more accurate measure of certainty, much as weak inducers in an ensemble can be combined to form a stronger classifier.

These multiple measures are incorporated into the strategy of Aggregate Certainty Estimators. With this technique, the votes of classifiers are weighted by certainty as determined on an instance-by-instance basis. Each classifier is trained using a different algorithm on the same training set data. Then each instance in the test set is assigned a class value and an overall certainty rating for that classification by each of n classifiers. Multiple factors are taken into consideration when determining this overall certainty rating. For example, six different certainty estimators are used to calculate certainty in the prediction of a decision tree classifier. A given instance would receive six certainty ratings, reflecting properties such as the purity of the leaf node in which it was classified and the number of instances classified at that leaf. These six numbers are then aggregated to produce an overall certainty rating for the decision tree's classification of this particular instance. A similar method is used to calculate an overall certainty rating for each of the classifiers. The class label assigned to the instance is then calculated by summing the weights for each possible label and selecting the class label with the maximum total.

The technique of Aggregate Certainty Estimators is shown to achieve higher average classification accuracy over 36 data sets than the standard combination strategies employed by Bagging and Boosting as well as the SelectBest strategy of allowing the most accurate classifier in the system to make all the classifications. It also outperforms Arbitration [Ortega et al., 2001] and the Stacking algorithm presented by Dzeroski and Zenko [2004] in terms of average classification accuracy.

Section two of this work provides an overview of related research. Section three outlines the Aggregate Certainty Estimators algorithm. Section four presents certainty estimators for five common classification algorithms. Section five provides results comparing Aggregate Certainty Estimators with standard voting, voting by accuracy, the SelectBest strategy, Arbitration, and Modified Stacking. Section six outlines conclusions and suggests options for further research.

7.2 Related Work

One common method of combining votes of base classifiers on an instance-by-instance basis is to use posterior class probabilities [Rokach, 2010]. However, most traditional classifiers are not naturally designed to output these probabilities. Thus, statistical or heuristic estimates of how certain a given classifier should be in its classification of an instance can prove useful.

For example, Provost and Domingos [2003] found that using a simple Laplace correction improves probability-based rankings. Specific pruning strategies and bagging techniques also resulted in ranking improvement. Ferri et al. [2003] found that the performance of these trees could further be improved by using different splitting criteria and a new smoothing technique that considers all the nodes along the classification path from root to leaf. These types of techniques have also been applied to rule-based classifiers. For example, Dzeroski et al. [1993] used the m-estimate to smooth probabilities and make predictions more effective.

While Naïve Bayes classifiers naturally produce a probability distribution for all class values, Domingos and Pazzani [1997] found that the power of the Naïve Bayes classifier lies more in the ordering of the classes than the actual probabilities predicted. They found that the classifier performs surprisingly well, even when the prior assumption of feature independence is clearly not met. Other researchers, He and Xiaoqing [2007], introduce a number of smoothing methods that can improve these probability estimates, resulting in more accurate and stable estimates than those that can be achieved with Laplace smoothing.

Unlike Bayesian models, multilayer perceptrons do not calculate class probabilities explicitly, but Ruck et al. [1990] provides a proof that the activation outputs of a multilayer perceptron approximate these probabilities. Richard and Lippman [1991] also found that the accuracy of these probability estimates depended on the network complexity, the amount of training data, and how well the training data represented true a priori class probabilities.

In addition, while combining votes of base inducers according to class probabilities may be an intuitive method, other methods benefit from taking additional factors into consideration. For example, Dolev et al. [2010] focus on attributes of the data set when determining how to weight votes. Their algorithm assumes a percentage of corrupted data, and they statistically analyze the data and attempt to remove corrupted data by identifying outliers in the distributions. The estimated distribution parameters are also used to determine the likelihood of feature values in the training set and a corresponding certainty level for leaf nodes in a decision tree where these training set instances are classified. Carney et al. [1999] focus on the variances among the distributions of base classifier output when determining how votes should be combined. Ali and Pazzani [1996] compare simple voting to three other evidence combination methods. "Bayesian Combination" approximates the optimal Bayes approach, taking into account both accuracy on training set data and posterior probabilities when weighting rule output. The "Distribution Summation" method takes into account the number of training instances covered by a given rule. “Likelihood Combination” takes both coverage and training set accuracy into account when assigning weights.

In most cases in literature, estimates of certainty are calculated using a single measure, and improvements are aimed at finding ways to smooth and improve this single measure’s accuracy. Certainty measures may take into account variables such as the inherent properties of a classifier, how classifiers behave in an overall system, or distributions of attributes in a data set. But given the variety of things that may be considered when estimating certainty of classification, it stands to reason that an algorithm could greatly benefit from taking multiple variables into account. The certainty estimators incorporated in this work include traditional class probability estimators, but also consider other features discussed by researchers such as data set coverage and distributions of base classifier outputs.

In order to demonstrate the usefulness of our proposed weighting system, we compare the technique of Aggregate Certainty estimators to other methods that weight votes of base classifiers on an instance-by-instance basis. Arbitration [Ortega et al., 2001] creates a “referee” to determine the certainty that a learning model has in its classification of the various subdomains of a given problem. Information about both the misclassification of instances and the classifiers themselves are used in the development of the meta-learner referees. Stacking [Wolpert, 1992] makes use of a meta-level learning algorithm that discovers the best way to combine outputs from the base level classifiers.

Dzeroski and Zenko [2004] found that the accuracy of an ensemble over a data set is often no better than the accuracy of one of the classifiers contributing to the ensemble. In order to justify the overhead of creating an ensemble, the ensemble should be able to perform better than a strategy of simply selecting the best classifier by cross-validation. In the algorithms Dzeroski and Zenko explore, only their Modified Stacking strategy was able to consistently achieve this level of performance. We demonstrate that our strategy of Aggregate Certainty Estimators also tends to outperform the single most accurate base classifier in a given ensemble.

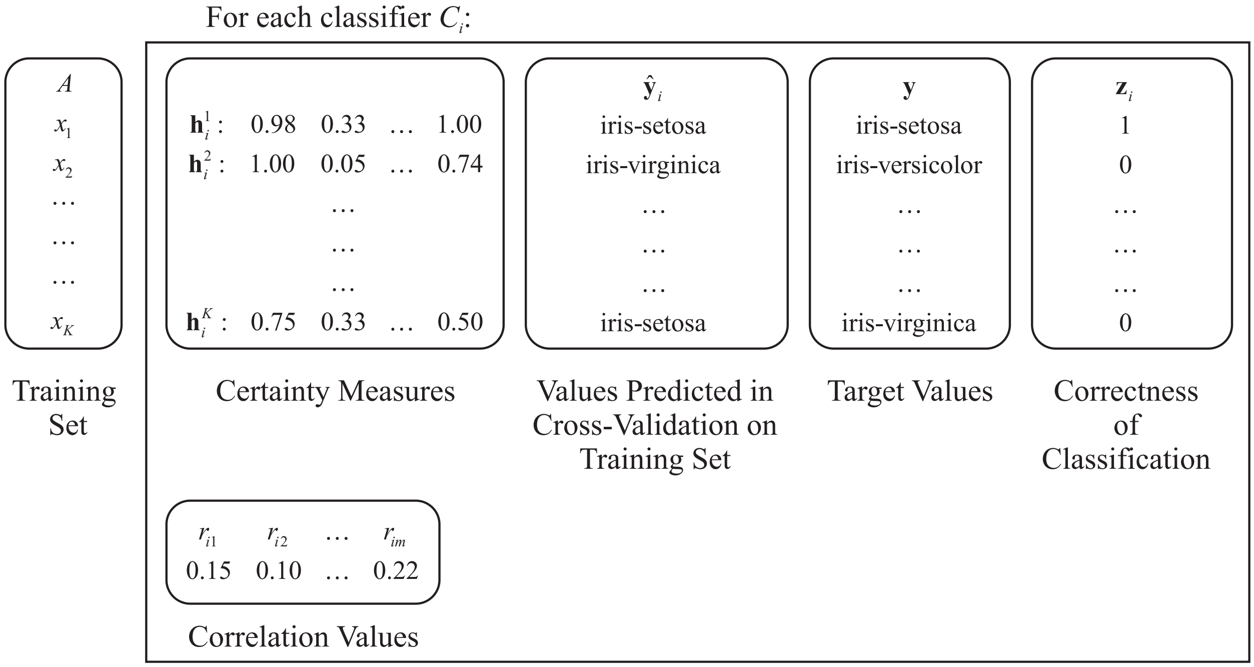

For each classifier C_i:
A
x_1
x_2
…
…
…
x_K
Training Set
$\mathbf{h}_i^1$: 0.98 0.33 … 1.00
$\mathbf{h}_i^2$: 1.00 0.05 … 0.74
…
…
…
$\mathbf{h}_i^K$: 0.75 0.33 … 0.50
Certainty Measures
$\hat{\mathbf{y}}_i$
iris-setosa
iris-virginica
…
…
…
iris-setosa
Values Predicted in Cross-Validation on Training Set
$\mathbf{y}$
iris-setosa
iris-versicolor
…
…
…
iris-virginica
Target Values
$\mathbf{z}_i$
1
0
…
…
…
0
Correctness of Classification
r_{i1} r_{i2} … r_{im}
0.15 0.10 … 0.22
Correlation Values

Figure 7.1: Values calculated for a component classifier during training of Aggregate Certainty Estimators

7.3 Aggregate Certainty Estimators

Let $C_1 ... C_n$ be classifiers constructed using instances from a training set A. For each classifier C_i, m pre-defined certainty estimators are used to calculate certainty estimates. For a given instance k in the training set, a vector $\mathbf{h}_i^k$ of m certainty values is calculated, with h_{ij}^k being the value assigned to instance k by the jth certainty estimator of classifier C_i.

For each classifier C_i, let $\hat{\mathbf{y}}_i$ be the predictions of C_i over the training set A, with $\hat{y}_i^k$ being the class label assigned by classifier C_i when instance k appeared in a test fold during cross-validation on the training set. Let $\mathbf{y}$ be a vector of the target values for the training instances, and $\mathbf{z}_i$ be a vector describing $\hat{\mathbf{y}}_i = \mathbf{y}$. In other words, if $\hat{y}_i^k = y^k$, then $z_i^k = 1$; if not, $z_i^k = 0$. For each classifier C_i, let $\mathbf{r}_i$ be a vector of correlation values, where r_{ij} is the correlation between $\mathbf{z}_i$ and $\mathbf{h}_{ij}$. The values in $\mathbf{r}_i$ are then scaled to sum to one. Figure 7.1 outlines the values calculated for each classifier in the ensemble.

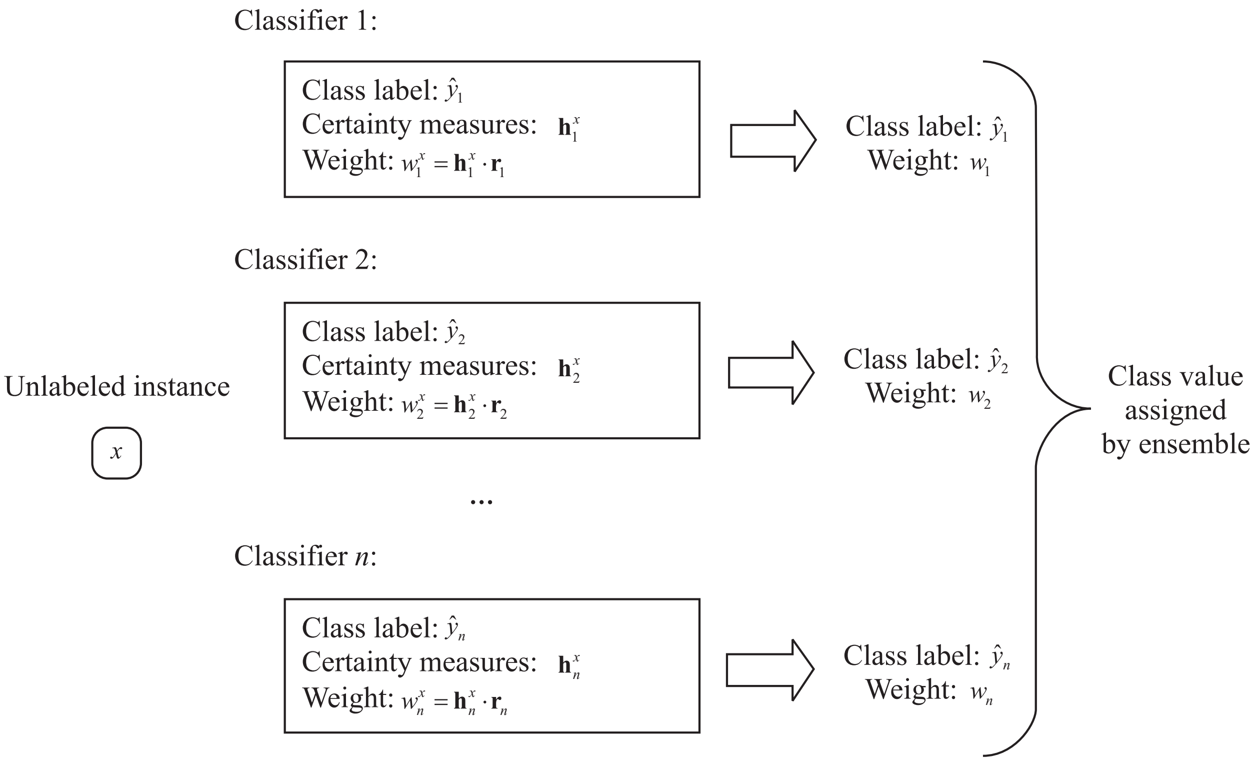


Figure 7.2: Values calculated by Aggregate Certainty Estimators to evaluate an unlabeled instance

For each unlabeled instance x, let $\hat{y}_i^x$ be the class label assigned to instance x by classifier C_i. Let $\mathbf{h}_i^x$ be a vector of certainty values calculated for the classification of instance x by classifier C_i. These values will be used in determining how much weight the overall ensemble should assign to the classification $\hat{y}_i^x$. In order to make the values assigned by the various estimators more uniform among the classifiers, h_{ij}^x is normalized using the maximum and minimum values from the vector $\mathbf{h}_{ij}$ of values calculated for training set instances.

Let w_i^x be the dot product of $\mathbf{h}_i^x$ and $\mathbf{r}_i$. This aggregate measure is then used to weight $\hat{y}_i^x$. The class label assigned to $\mathbf{x}$ by the overall ensemble is calculated by summing the weights for each possible label and selecting the class label with the maximum total. Figure 7.2 outlines the values calculated for unseen instances. The strategy of Aggregate Certainty Estimators is outlined in Figure 7.3.

```
1.Train each of $n$ classifiers $C_1...C_n$ using training set $A$.
    A. Determine the following for each classifier $C_i$:
        1. For each instance $k$ in $A$:
            a. Calculate vector $\mathbf{h}_i^k$ of certainty values using $m$ estimators specific to $C_i$
            b. Calculate $\hat{y}_i^k$, the prediction of class label by $C_i$ when $k$ appeared in a test fold
               during cross-validation on $A$
            c. Identify $y^k$, the target value for instance $k$
        2. Define $\mathbf{z}_i$ to be a vector describing $\hat{\mathbf{y}}_i = \mathbf{y}$
        3. Calculate vector $\mathbf{r}_i$ of correlation values where $r_{ij}$ is the correlation
           between $\mathbf{z}_{ij}$ and $\mathbf{h}_{ij}$ (Scale each value in $\mathbf{r}_i$: $r_{ij} = r_{ij}/\Sigma_{j=1}^{m} r_{ij}$ so that values sum to one)
2. For an unlabeled instance $x$:
    A. For each classifier $C_i$:
        1. Determine $\hat{y}_i^x$, the class value of $\mathbf{x}$ as predicted by $C_i$
        2. Create vector $\mathbf{h}_i^x$ of certainty values using $m$ estimators specific to $C_i$
           (Scale each value in $\mathbf{h}_i^x$: $h_{ij}^x = (h_{ij}^x - min_{ij})/(max_{ij} - min_{ij})$ where
           $max_{ij}$ and $min_{ij}$ are the maximum and minimum $h_{ij}^k$ values from the training set)
        3. $w_i^x = \mathbf{h}_i^x \cdot \mathbf{r}_i$
    B. Class value for $\mathbf{x} = \text{argmax}_{y \in Y}(\Sigma_{i=1}^{n} \delta(y, \hat{y}_i^x) w_i^x)$
        $\delta(y, \hat{y}_i) = \begin{cases} 1 \text{ when } \hat{y}_i = y \\ 0 \text{ otherwise} \end{cases}$
```

Figure 7.3: Aggregate Certainty Estimators

As concrete example, six measures were used to describe certainty of classification by a rule-based classifier in our experiments. These measures include such factors as the purity of classification of training set instances covered by the rule and the number of instances covered. These six numbers were then averaged to produce an overall certainty measure. In a similar fashion, overall certainty measures are calculated for each of the five learning algorithms incorporated in the multi-classifier system. These measurements are calculated for both training set and test set instances.

This means that, for a training set with 135 instances (e.g. the iris data set using ten-fold cross-validation), a 135 X 5 matrix would be generated, with one row for every instance in the training set and one column for every classifier incorporated in the overall system. A correlation value would then be calculated between each column of the matrix and a column of “1”s and “0”s that described whether each of the training set instances was correctly classified in cross-validation experiments on the training set (e.g. train a rule-based classifier on 134 test set instances and determine if the resulting classifier could correctly label the 135th instance of the training set[1]).

In our example, five certainty measures would also be calculated for each test set instance. Each measure would then be multiplied by the correlation value for its respective classifier. These values would then be used to weight the predictions from each classifier. For example, assume the following for a given instance:

	Classification	Certainty Estimator	Correlation Value
Decision Tree	iris-setosa	0.93	0.15
Rule-based Classifier	iris-setosa	0.84	0.10
Instance-based Classifier	iris-virginica	0.23	0.36
Naive Bayes Classifier	iris-virginica	0.87	0.45
Multilayer Perceptron	iris-setosa	0.66	0.22

Summing the votes for iris-setosa: $0.93 * 0.15 + 0.84 * 0.10 + 0.66 * 0.22 = 0.3687$. Summing the votes for iris-virginica: $0.23*0.36+0.87*0.45 = 0.4743$. Finding the maximum value, the overall system would assign the label of "iris-virginica" to this particular instance.

7.4 Multiple Certainty Estimators

This section contains the information about the certainty estimators used to predict certainty in classifications for each of five different algorithms. The five algorithms used in this work were selected because they are representative of standard classes of algorithms used in machine learning. Many of the certainty estimators presented here could be adapted for use with similar machine learning algorithms. The algorithms used in this work are implemented using Weka open source code [Witten and Frank, 2005]. Default settings are used

[1] In the experiments described in this paper, hold-one-out cross-validation was used to generate this column, simply to provide a higher degree of accuracy in evaluating the measures. In practice, cross-validation with a low number of folds could generate this column at much less expense in terms of computation time.

for each of the algorithms to allow for easier reproduction of results. These settings allowed for reasonable performance of the base classifiers on the test data sets.

While we have tried to select diverse models to represent the spectrum of machine learning algorithms, the technique of Aggregate Certainty Estimators could be applied to ensembles with any number and type of base-level classifiers. The systems of the five base-level classifiers discussed here are designed simply to present the concept. One classifier of each type is used for parsimony and to avoid skewing the ensemble in favor of any particular classifier.

Efficacy of the various certainty estimators is evaluated using 36 data sets taken from the UCI Repository [Hettich et al., 1998]. Table 7.1 provides information about these data sets. Data sets were selected so as to achieve variety in number of instances, attributes, attribute types, and output classes. The data sets range from 90 to 2310 instances, 5 to 70 attributes, and 2 to 24 output classes. Roughly a third of the data sets feature discrete attributes, another third have real-valued attributes, and the remaining data sets have a mixture of discrete and real-valued attributes. Ten of the data sets contain missing values. In the case of discrete attributes, missing values were replaced by the most common value for the given attribute. For data sets with real-valued attributes, unknown values were replaced with the average value for the attribute.

To evaluate the certainty measures for each of the algorithms studied, ten-fold cross-validation experiments were conducted for each of the data sets. Instances were marked as correctly or incorrectly classified based on the classifier's ability to classify the instance when it appeared in the test set. This correctness of classification is compared to the certainty measure assigned to each instance by each of the certainty estimators. Please note that the correct/incorrect labels assigned to test instances and used for the purposes of evaluating the certainty measures are independent of the correct/incorrect labels assigned to instances during training.

Table 7.1: Information for Data Sets

	Number of Instances	Number of Attributes	Output Classes	Attribute Type	Missing Attributes?
anneal	898	39	6	mixed	no
audiology	226	70	24	discrete	yes
balance-scale	625	5	3	real	no
bupa	345	7	2	real	no
car	1728	7	4	discrete	no
cmc	1473	10	3	mixed	no
colic	368	23	2	mixed	yes
credit-a	690	16	2	mixed	yes
credit-g	1000	21	2	mixed	no
dermatology	366	35	6	mixed	yes
diabetes	768	9	2	real	no
ecoli-c	336	8	8	real	no
glass	214	10	7	real	no
haberman	306	4	2	real	no
heart-disease	294	14	5	mixed	yes
heart-statlog	270	14	2	real	no
hepatitis	155	20	2	mixed	yes
ionosphere	351	35	2	real	no
iris	150	5	3	real	no
lymph	148	19	4	mixed	no
monks	432	7	2	discrete	no
postop	90	9	3	discrete	no
primary-tumor	339	18	22	discrete	yes
segment	2310	20	7	real	no
sonar	208	61	2	real	no
soybean	683	36	19	discrete	yes
spect	267	23	2	discrete	no
tic-tac-toe	958	10	2	discrete	no
vehicle	846	19	4	real	no
vote	461	17	2	discrete	no
vowel	990	14	11	mixed	no
wine	178	14	3	real	no
wisconsin-cancer	286	10	2	discrete	yes
yeast	1484	9	10	real	no
yugoslavia-cancer	699	10	2	real	no
zoo	101	17	7	discrete	no

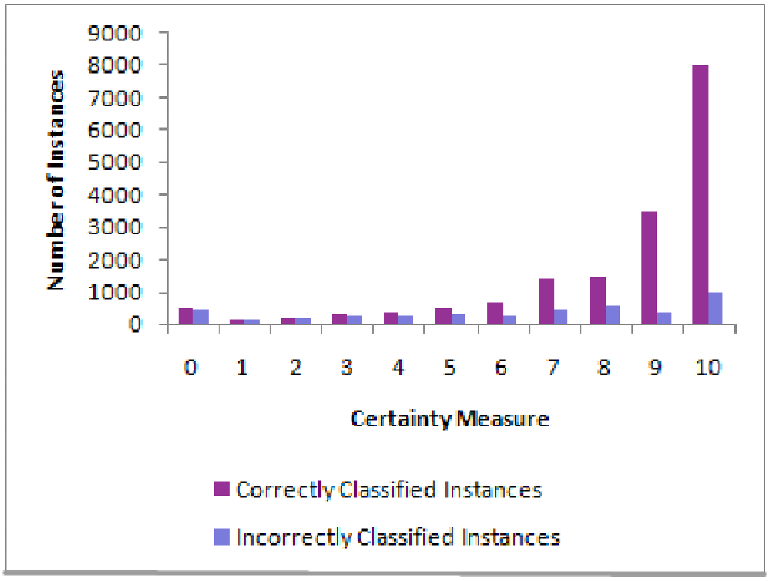


Figure 7.4: Decision Tree Purity Certainty Estimator - Treatment of Correct and Incorrect Instances

Just as base inducers must exhibit at least slightly better-than-random performance in order to provide benefit to an ensemble, we specify a baseline measure of performance for our certainty estimators.

Graphs such as the one in figure 7.4 were constructed for each of the measures studied. This graph shows the number of instances receiving a given certainty value that were correctly and incorrectly classified. While real-valued, unbinned certainty estimates are used in the actual classification experiments, for clarity in graphing, certainties are grouped in discrete bins (e.g. certainty values from 0.5 to 0.59 are all graphed as 0.5). The bar on the left for each bin represents the number of instances receiving this certainty value that were correctly classified. The bar on the right represents the number of instances that were incorrectly classified. For the purity certainty estimator, the far right-hand bin in the graph shows that, out of all 36 data sets, 5128 instances receiving a certainty value of 1.0 were correctly classified, and 863 instances receiving this certainty value were incorrectly classified.

We fit a trend line to the percentage of correctly classified instances in each of the various bins using least squares regression. Each measure included in our experiments sat-

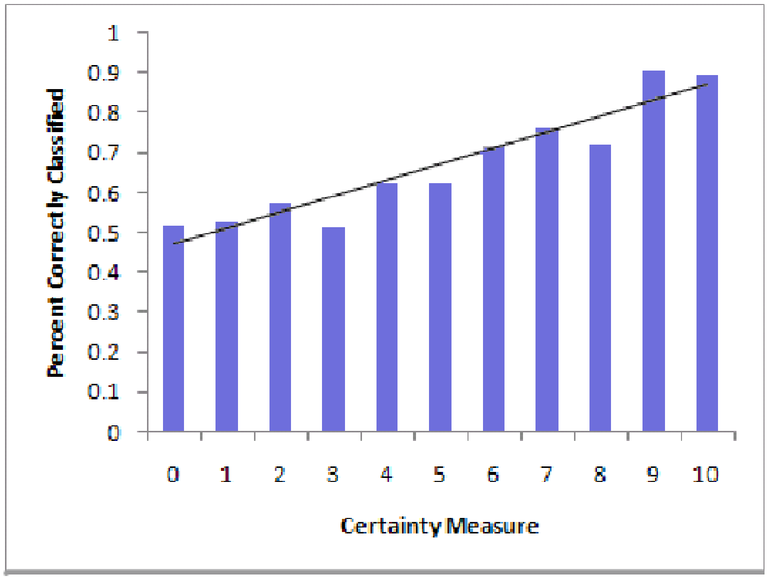


Figure 7.5: Decision Tree Purity Certainty Estimator - Percentage of Correctly Classified Instances Per Bin

isfies the minimum requirement that this trend line have a positive slope. Intuitively this indicates that the measure is more likely to assign a high certainty value to a correctly classified instance as opposed to an incorrectly classified one. Figure 7.5 shows such a line for the decision tree purity certainty estimator.

Each subsection contains information about the various algorithms and the certainty estimators used for each. A table outlining the correlation between the various certainty estimators and correctness in test set classification is also included.

7.4.1 Decision Tree - J48

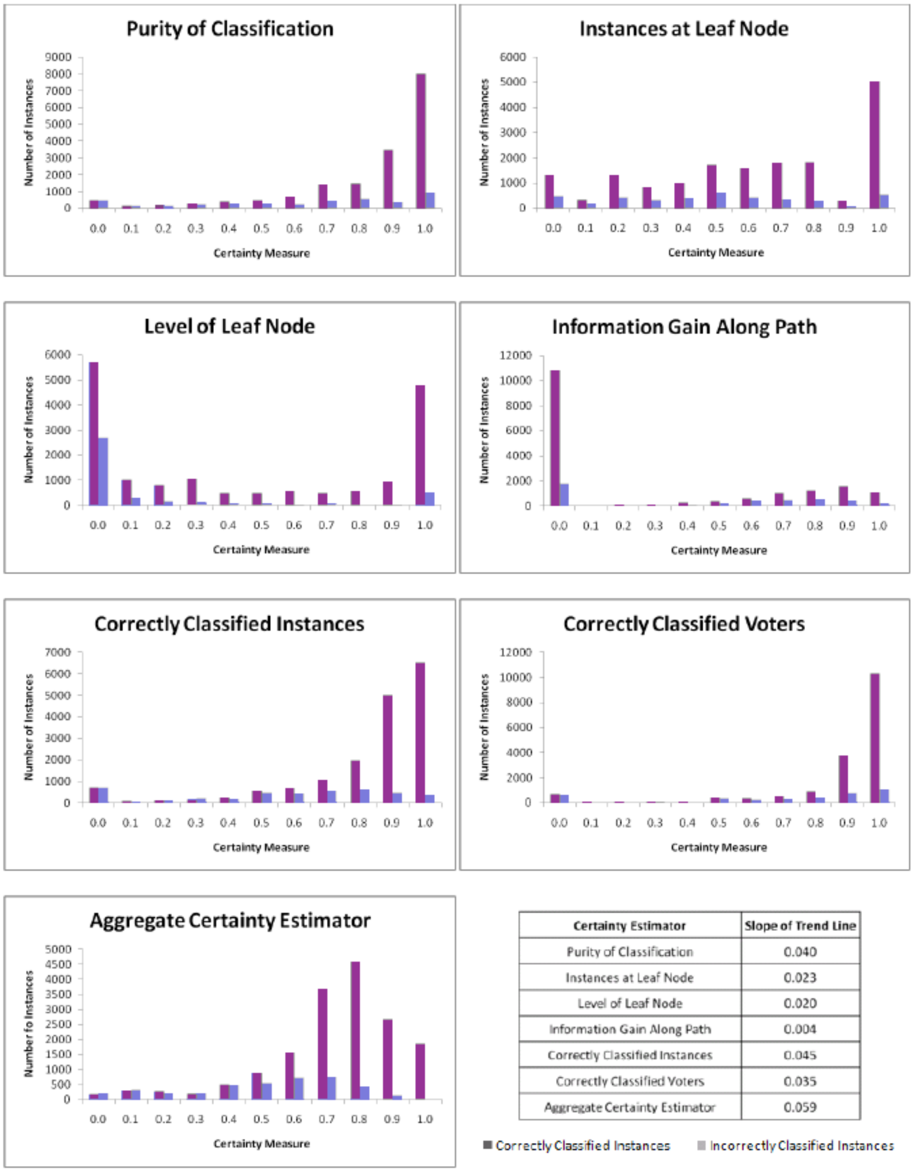
To demonstrate the effectiveness of multiple certainty measures, we provide a case study with the measures used for Decision Trees. The J48 algorithm is the Weka implementation of the C4.5 algorithm [Quinlan, 1993], an extension of the ID3 decision tree algorithm [Quinlan, 1986]. Six different certainty estimators are used to predict certainty in this algorithm's classification of a given instance:[2]

[2] For a more rigorous presentation of these and all other certainty estimators mentioned in this paper, please see the Appendix.

1. The number of instances with the predicted class at the leaf node when the given instance is classified (the purity of classification at that node)
2. The number of instances at the leaf node
3. The level of the tree at which the given instance is classified
4. The average of the information gain statistics along the classification path (normalized by maximum possible information gain for a given data set)
5. The percentage of instances at the leaf node that were correctly classified in hold-one-out cross-validation experiments on the training set
6. The percentage of instances at the leaf nodes with the predicted class for that node that are correctly classified in hold-one-out cross validation on the training set

The first certainty estimator is a standard method for predicting certainty in the classification of a decision tree [Witten and Frank, 2005]. The second and third provide an effective complement to the first by providing information about the amount of overfit and thus how much the first should be trusted. The fourth certainty estimator provides information about how effectively a given attribute is able to split the data at each level of the decision tree, assuming that strong attributes will lead to more confident classifications. The fifth identifies how effective the classifier is at classifying the instances in this particular section of the data. The sixth certainty estimator provides information about how effectively the classifier was able to classify the instances specifically contributing to the classification of the given instance.

Figure 7.6 provides information about the behavior of each certainty estimator on the data sets shown in Table 7.1. For example, the graph in the top left of the figure displays information about the certainty estimator measuring purity of classification at the leaf node of a decision tree.



Certainty Estimator	Slope of Trend Line
Purity of Classification	0.040
Instances at Leaf Node	0.023
Level of Leaf Node	0.020
Information Gain Along Path	0.004
Correctly Classified Instances	0.045
Correctly Classified Voters	0.035
Aggregate Certainty Estimator	0.059

Figure 7.6: Decision Tree Certainty Estimators Treatment of Correct and Incorrect Instances

Table 7.2 shows the correlation between correctness of classification and values provided by each certainty estimator. One can infer from this and Figure 7.6 that the purity of classification measure and the two certainty measures concerned with correctly classified instances appear to be better predictors of correctness of classification for test instances. However, the other certainty estimators do provide additional information that may be useful, particularly when taken into consideration with the more accurate certainty-predicting measures.

For example, with the haberman data set, a majority of the correctly classified instances were assigned a certainty rating of 0.82 by the purity certainty estimator. The few instances receiving higher certainty ratings were all misclassified. However, the misclassified instances that received deceptively high certainty ratings from the purity certainty estimator were generally found in leaf nodes that contained only a few instances. Thus, they received lower certainty ratings both from the certainty estimator that measured the percentage of instances at the leaf node and the one that measured the level of the tree. A combination of these certainty estimators is better at predicting whether or not an instance from this data set will be correctly classified.

Table 7.2 shows that, on average, the certainty estimator relating to information gain was slightly negatively correlated with correctness of classification. However, this measure proved effective on select data sets. For example, on the hepatitis data set, correlation between the information gain certainty estimator and correctness of training set classification was 0.153, while correlation with the purity of classification estimator was only 0.091. Overall accuracy of the Aggregate Certainty Estimators strategy on the hepatitis data set as evaluated by ten-fold cross-validation was 85.16% when the information gain estimator was included, and only 84.52% when it was excluded. The inclusion of this certainty estimator resulted in an improvement in accuracy of Aggregate Certainty Estimators on 16 of the 36 data sets studied. It reduced accuracy on only three of the data sets.

Table 7.2: Decision Tree Certainty Estimators and Correlation with Correctly Classified Instances

Certainty Estimator	Correlation
1. Purity of Classification	0.219
2. Instances at Leaf Node	0.167
3. Level of Leaf Node	0.199
4. Information Gain Along Path	-0.072
5. Correctly Classified Instances	0.280
6. Correctly Classified Voters	0.248
Aggregate Certainty Estimator	**0.292**

Similar patterns can be seen for the certainty estimators presented for all of the algorithms studied. For each of the estimators studied, higher certainty values generally corresponded with a higher percentage of correctly classified instances. More specifically, a trend line fit to a percentage of correct instances for each binned certainty estimator had a positive slope (e.g. Figure 7.5); instances assigned the highest certainty measure were more likely to be correctly classified than instances assigned the lowest certainty measure for each of the estimators studied. This is true even for certainty estimators that exhibit low average correlation with correctness of classification. However, in each case, the aggregate certainty estimator was significantly more correlated with correctness of classification than each of the individual estimators.

The Friedman test indicates that there are significant differences among the correlations of the various certainty measures. ($92.45 \sim \chi^2, DF = 6, p <= 0.0001$). The Bonferroni-Dunn post-hoc test indicates that the differences in average ranks between aggregate certainty estimator and five of the six other estimators exceeds the critical difference for significance at a certainty level of 95% (Adjusted $\alpha = \frac{0.05}{7-1}$, Critical difference = 1.319, Mean rank differences: $2.750, 2.438, 1.921, 4.781, 0.953, 2.031$).

7.4.2 Multilayer Perceptron trained with Backpropagation

In contrast, we also present an explanation of the certainty estimators used for the multilayer perceptron. One of the most common methods of training a multilayer perceptron, backpropagation incrementally changes the weights between nodes when these weights are responsible for the misclassification of instances during training [Rumelhart et al., 1986]. These experiments use a multilayer perceptron with a single hidden. The following are considered in trying to predict certainty in classification by the Multilayer Perceptron:

1. The activation output for the selected classification
2. The difference between the highest and second highest activation outputs
3. The percentage of the five neighbors nearest in activation output that were correctly classified in hold-one-out cross-validation on the training set
4. The percentage of the five neighbors nearest in activation output of the hidden layer that were correctly classified in hold-one-out cross-validation on the training set
5. The average difference in activation output between the selected classification and its five nearest neighbors compared to the average of this statistic computed for all instances
6. The average difference in hidden-layer activation output between the selected classification and its five nearest neighbors compared to the average of this statistic computed for all instances

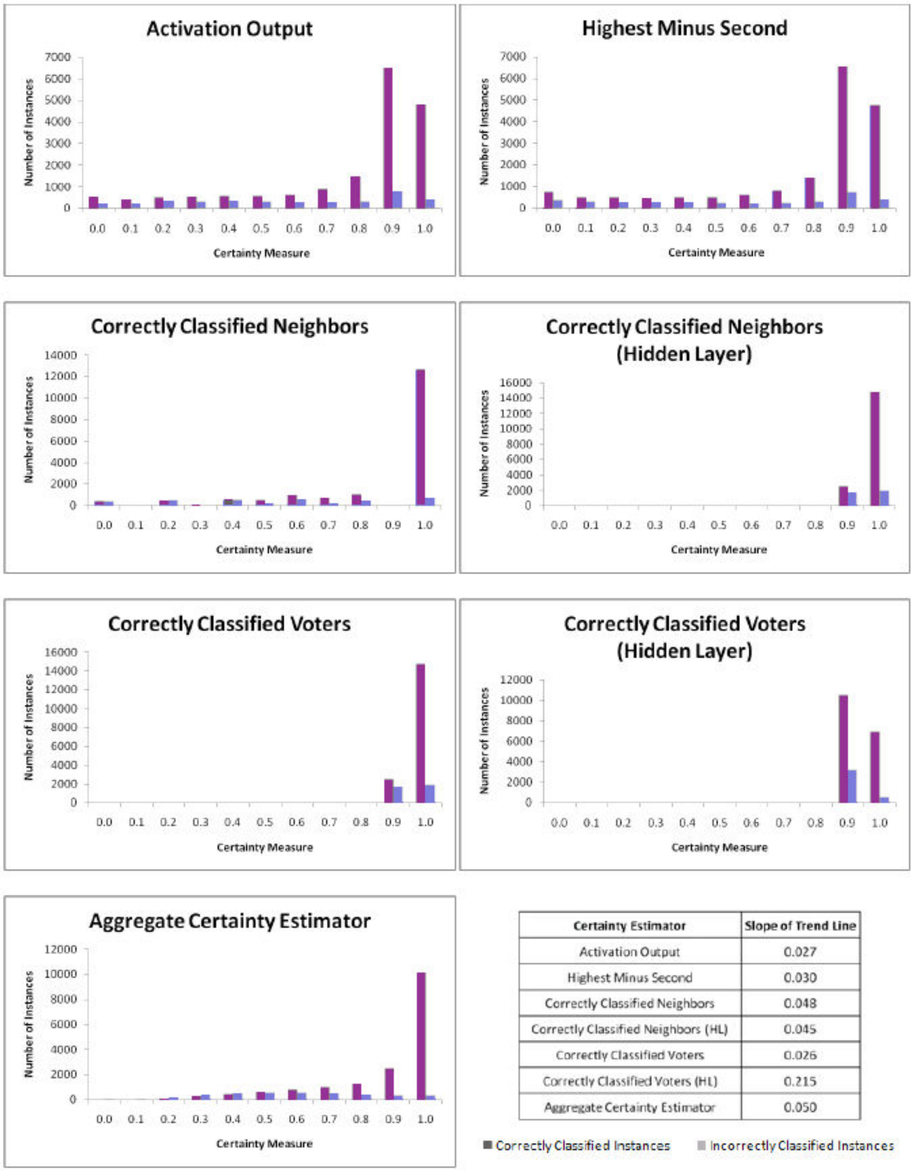
The first and second certainty estimators provide information about the certainty of a given classification and certainty relative to other possible classifications. The third and fourth provide information about how confident the classifier is on this region of the input space. All the instances in the training set are considered, and the five with output vector most similar to the instance in question are then used to calculate the certainty estimator. The third certainty estimator uses the outputs from the standard output nodes to identify the nearest neighbors. The fourth certainty estimator uses the outputs from the hidden nodes. The fifth and sixth certainty estimators provide information about how similar a given instance is to previously seen instances, based on the assumption that the classifier will be more effective at predicting a class value for an instance similar to one that it has seen before. Figure 7.7 shows the behavior of these certainty measures on data set instances.

Table 7.3: Multilayer Perceptron Certainty Estimators and Correlation with Correctly Classified Instances

Certainty Estimator	Correlation
1. Activation Output	0.053
2. Highest Minus Second	0.051
3. Correctly Classified Neighbors	0.295
4. Correctly Classified Neighbors (Hidden Layer)	0.266
5. Average Distance to Neighbors	0.239
6. Average Distance to Neighbors (Hidden Layer)	0.157
Aggregate Certainty Estimator	**0.310**

As illustrated by the graphs in Figure 7.7, all of these heuristics tend to assign a 1.0 certainty rating to a large number of correctly classified instances. The number of correctly classified instances at each certainty rating tends to taper off as the ratings become lower. On average, the heuristics for this classifier were more highly correlated with each other than the heuristics for other classifiers. However, an examination of the certainty ratings assigned to individual instances in the data sets shows that there is enough variation that each heuristic does provide some extra information to a classifier. The biggest jump in correlation with correctness of classification between individual certainty measures and an aggregate certainty estimator was seen with the Multilayer Perceptron.

Table 7.3 reports how values assigned by these certainty measures correlate with correctness of classification. The Friedman test indicates that there are significant differences among the correlations of the various certainty measures. ($127.99 \sim \chi^2, DF = 6, p <= 0.0001$). The Bonferroni-Dunn post-hoc test indicates that the differences in average ranks between aggregate certainty estimator and five of the six other estimators exceeds the critical difference for significance at a certainty level of 95% (Adjusted $\alpha = \frac{0.05}{7-1}$, Critical difference $= 1.216$, Mean rank differences: $4.329, 4.529, 0.943, 1.643, 1.871, 2.786$).



Certainty Estimator	Slope of Trend Line
Activation Output	0.027
Highest Minus Second	0.030
Correctly Classified Neighbors	0.048
Correctly Classified Neighbors (HL)	0.045
Correctly Classified Voters	0.026
Correctly Classified Voters (HL)	0.215
Aggregate Certainty Estimator	0.050

■ Correctly Classified Instances ■ Incorrectly Classified Instances

Figure 7.7: Multilayer Perceptron Certainty Estimators Treatment of Correct and Incorrect Instances

7.4.3 Rule-Based Classifier - Decision Table

These experiments use one of Weka's rule-based classifiers called a Decision Table [Kohavi, 1995]. This algorithm selects a set of attributes to be used in determining classification, and produces a classification for each combination of observed values for these attributes. The following attributes are taken into consideration when trying to predict certainty in this algorithm's classification of a given instance:

1. The number of instances with the predicted class covered by the rule
2. The number of antecedents in the rule
3. The number of instances covered by the rule
4. The percentage of instances covered by the rule that were correctly classified in hold-one-out cross-validation experiments on the training set
5. The percentage of instances covered by the rule with the predicted class for that rule that were correctly classified in hold-one-out cross-validation on the training set
6. Whether or not this instance is covered by a rule

The rationale for these certainty estimators is similar to the rationale for the decision tree certainty estimators. The first is a standard measure of certainty. The second and third assess the probability of overfit or underfit. The fourth and fifth measure the effectiveness and strength of classification. They indicate how effectively the decision table was able to classify instances that would end up in this region and how effectively the most pertinent instances in this region can be classified. The sixth certainty estimator indicates whether or not a rule was found in the table that covered the given instance to be classified. Table 7.4 shows how values assigned by these certainty measures correlate with correctness of classification.

The Friedman test indicates that there are significant differences among the correlations of the various certainty measures. ($55.96 \sim \chi^2, DF = 6, p <= 0.0001$). The Bonferroni-Dunn post-hoc test indicates that the differences in average ranks between aggregate certainty estimator and all six other estimators exceeds the critical difference for significance at a certainty level of 95% (Adjusted $\alpha = \frac{0.05}{7-1}$, Critical difference = 1.523, Mean rank differences: $1.917, 4.542, 2.583, 2.208, 2.874, 2.208$).

Table 7.4: Rule-Based Classifier Certainty Estimators and Correlation with Correctly Classified Instances

Certainty Estimator	Correlation
1. Purity of Classification	0.147
2. Number of Antecedents	-0.004
3. Number of Instances Covered	0.110
4. Correctly Classified Instances	0.139
5. Correctly Classified Voters	0.102
6. Instance is Covered by Rule	0.217
Aggregate Certainty Estimator	**0.240**

7.4.4 Instance-Based Classifier

With the instance-based k-nearest-neighbor algorithm, an instance is classified based on the classifications of the k instances nearest that instance [Cover and Hart, 1967]. These experiments use the 5-nearest-neighbor version of the algorithm. Attribute values are normalized, and standard rather than distance weighted voting is used. Six different options are used to predict certainty in this algorithm's classification of a given instance:

1. The percentage of the first five neighbors that have the same classification as the predicted class for those five neighbors
2. The difference between the distance weighted vote of the predicted class and the distance weighted vote of the next highest class
3. The average distance from this instance to its first five neighbors (normalized and subtracted from one)
4. The percentage of the first five neighbors that were correctly classified in hold-one-out cross-validation on the training set
5. The percentage of neighbors with the predicted class that were correctly classified in hold-one-out cross-validation on the training set
6. A comparison of 3-NN, 5-NN and 7-NN classifications of a given instance

The first and second certainty estimators indicate the general certainty in a classification, and how confident that classification is relative to other possible classifications. The third measures how close the neighbors are to the individual instance, making the assumption that an instance closer to other instances is more likely to be correctly classified. The fourth and fifth certainty estimators measure the classification accuracy of instances in this

Table 7.5: Instance-Based Classifier Certainty Estimators and Correlation with Correctly Classified Instances

Certainty Estimator	Correlation
1. Neighbors in Agreement	0.325
2. Highest Minus Second	0.343
3. Average Distance to Neighbors	0.114
4. Correctly Classified Neighbors	0.276
5. Correctly Classified Voters	0.198
6. 3-NN vs. 5-NN vs. 7-NN	0.242
Aggregate Certainty Estimator	**0.358**

region and the accuracy on instances contributing to the classification of the instance in question. The last certainty estimator indicates the effectiveness of using this particular number of neighbors to classify the given instance. Table 7.5 reports correlation between values assigned by these measures and correctness of classification.

The Friedman test indicates that there are significant differences among the correlations of the various certainty measures. $(99.96 \sim \chi^2, DF = 6, p <= 0.0001)$. The Bonferroni-Dunn post-hoc test indicates that the differences in average ranks between aggregate certainty estimator and four of the six other estimators exceeds the critical difference for significance at a certainty level of 95% (Adjusted $\alpha = \frac{0.05}{7-1}$, Critical difference $= 1.261$, Mean rank differences: $1.257, 0.571, 4.114, 2.143, 3.143, 2.971$).

7.4.5 Naïve Bayes Classifier

The Naïve Bayes classifier uses Bayesian logic to predict class values for each instance based on the probabilities of the attribute values for that instance [Lang, 1995] [Mitchell, 1997]. The following are considered when trying to predict certainty in classification of a given instance by the Naïve Bayes classifier:

1. Probability of the class value predicted by the Naïve Bayes classifier
2. The distance between the predicted probability and the probability of the second most likely class value for the instance

Table 7.6: Naïve Bayes Classifier Certainty Estimators and Correlation with Correctly Classified Instances

Certainty Estimator	Correlation
1. Probability of Class Value	0.303
2. Highest Minus Second	0.298
3. Highest Minus Remaining	0.303
4. Value Probability Averages	0.075
5. Correctly Classified Neighbors	0.371
6. Correctly Classified Voters	0.306
Aggregate Certainty Estimator	**0.394**

3. The distance between the predicted probability and the sum of the probabilities for the remaining class values
4. The average probability across the data set of each attribute value in the instance
5. The percentage of the five neighbors nearest in probability that were correctly classified in hold-one-out cross-validation on the training set
6. The percentage of the nearest five neighbors with the same class value as this instance that were correctly classified in hold-one-out cross-validation on the training set

The first certainty estimator is used because it is the standard way of predicting the certainty of a Naïve Bayes classifier. The second and third certainty estimators are attempts to gain more information about how confident the classifier is in its ordering. The fourth certainty estimator addresses the fact that attribute values with lower representation in a data set may be less effective at contributing to a correct classification. The fifth certainty estimator is aimed at determining how confident the classifier is in this region of the input space. With this certainty estimator, the output probabilities of all the instances in the training data are taken into consideration. The five instances with output probabilities closest to those of the instance in question are then located, and the certainty estimate is calculated by observing the percentage of these five instances that were correctly classified in hold-one-out cross-validation on the training set. The sixth certainty estimator focuses specifically on neighbors with the same classification as the given instance. Table 7.6 shows how values assigned by these certainty measures correlate with correctness of classification.

The Friedman test indicates that there are significant differences among the correlations of the various certainty measures. ($56.59 \sim \chi^2, DF = 6, p <= 0.0001$). The Bonferroni-Dunn post-hoc test indicates that the differences in average ranks between aggregate certainty estimator and five of the six other estimators exceeds the critical difference for significance at a certainty level of 95% (Adjusted $\alpha = \frac{0.05}{7-1}$, Critical difference = 1.244, Mean rank differences: $1.542, 1.917, 1.417, 3.528, 0.777, 2.097$).

7.5 Results and Discussion

In this section, the technique of Aggregate Certainty Estimators is compared with a number of different ensemble combining strategies. Overall accuracy for each method is calculated by using ten-fold cross validation and averaging accuracy over each of the ten folds. Two sets of experiments are conducted. The first set demonstrates advantages of the aggregate certainty estimators. The strategy of Aggregate Certainty Estimators is shown to be more effective than the strategy of weighting by single certainty estimators. The second set of experiments compares Aggregate Certainty Estimators to a number of baseline ensemble creation strategies. Specifically, Aggregate Certainty Estimators are also shown to be competitive with other vote weighting strategies, the SelectBest method, and the methods of Arbitration and Modified Stacking.

7.5.1 Results

In order to motivate the need for multiple certainty estimators, the accuracies of different ensembles created with single certainty estimators are tested. Three different options are given for selecting single certainty estimators. The first alternate ensemble is created by using certainty estimators traditionally used in predicting certainty in a classification:

* Decision Tree: Purity of Classification
* Rule-Based Classifier: Purity of Classification
* Instance-Based Classifier: Neighbors in Agreement

* Naïve Bayes Classifier: Probability of Class Value

* Multilayer Perceptron: Activation Output

The next ensemble is also constructed using single certainty estimators to predict certainty. But in this case, an attempt is made to select more effective certainty estimators. For this ensemble, each algorithm uses the measure most highly correlated with whether or not an instance was correctly classified as the method for predicting certainty:

* Decision Tree: Correctly Classified Instances

* Rule-Based Classifier: Instance is Covered by Rule

* Instance-Based Classifier: Highest Minus Second

* Naïve Bayes Classifier: Correctly Classified Neighbors

* Multilayer Perceptron: Correctly Classified Neighbors

In addition, the Weka source code provides a way of calculating a probability distribution over possible classes for each instance classification. In most cases, Weka's method of calculating the probability distribution is similar to the first certainty estimator for each classifier presented in this work. Weka makes a few modifications and refinements to these measures. For a third comparison ensemble, the probability distributions predicted by Weka are used to weight the votes of each of the five classifiers.

All these techniques are then compared to the Aggregate Certainty Estimators' strategy of using a larger set of measures to predict certainty. The resulting predictive accuracies, shown in Table 7.7, demonstrate the utility of using more certainty estimators.

An application of the Friedman test reports significant differences in accuracy among the classifiers. $(14.19 \sim \chi^2, DF = 3, p <= 0.003)$. The Bonferroni-Dunn post-hoc test reveals that the differences in average ranks between Aggregate Certainty Estimators and two of the three other methods exceeds the critical difference for significance at a certainty level of 95% (Adjusted $\alpha = \frac{0.05}{4-1}$, Critical difference = 0.89, Mean rank differences: $1.01, 0.85, 0.92$). An algorithm-by-algorithm comparison between the various strategies using the Wilcoxon signed-rank test shows that Aggregate Certainty Estimators outperform the other three algorithms at a certainty level of 99% (p values<=0.0001, 0.001, 0.001).

In the next set of experiments, Aggregate Certainty Estimators is compared with several different baseline methods. Once again, overall accuracy for each method is calculated by using ten-fold cross validation and averaging accuracy over each of the ten folds. The first is a standard voting method where each classifier in an ensemble votes on the classification of an instance and the votes are weighted equally. The second baseline method weights the votes by the overall accuracy of the classifier on the training data for a given fold of the experiments. The third baseline method, identified here as the SelectBest method, chooses the classifier in the ensemble that achieved the highest accuracy on the training data and uses that classifier alone on the test data.

Aggregate Certainty Estimators is also compared to the method of Stacking found to be most effective by Dzeroski and Zenko [2004]. In this method, identified as Modified Stacking in the following analyses, the output probabilities of each of the component classifiers are given as input to a set of model trees. Each tree is designed to make a binary decision about a given possible output class, and the ensemble assigns a value to the instance according to which model tree has the highest positive certainty in its prediction. Table 7.8 shows the results of these comparisons.

Again the Friedman test reports significant differences in accuracy among the classifiers. ($15.51 \sim \chi^2, DF = 5, p <= 0.008$). An application of the Bonferroni-Dunn post-hoc test reveals that the differences in average ranks between Aggregate Certainty Estimators and four of the remaining five methods exceeds the critical difference for significance at a certainty level of 95% (Adjusted $\alpha = \frac{0.05}{6-1}$ Critical difference = 1.14, Mean rank differences: $1.21, 0.88, 1.24, 1.31, 1.43$). An algorithm-by-algorithm comparison between the various strategies using the Wilcoxon signed-rank test shows that Aggregate Certainty Estimators outperform the other five algorithms at a certainty level of 99% (p values<=0.001, 0.003, 0.014, 0.003, 0.008).

Table 7.7: Comparison of Predictive Accuracies of Aggregate Certainty Estimators with Single Certainty Measure Ensembles

Data Set	Traditional Certainty Estimators	Highest Correlation Estimators	Weka Outputs	Aggregate Confidence Ensembles
anneal	99.44	99.33	99.22	99.33
audiology	79.20	80.53	80.09	78.32
balance-scale	88.00	89.76	90.08	89.92
bupa	70.44	68.70	69.28	71.01
car	97.28	96.53	96.41	97.74
cmc	54.04	52.21	53.43	53.50
colic	83.97	84.51	83.97	84.24
credit-a	85.07	85.22	85.65	86.23
credit-g	75.60	74.60	75.20	75.40
dermatology	97.81	97.81	97.81	97.27
diabetes	76.43	76.43	76.56	76.56
ecoli-c	85.71	87.20	87.20	87.50
glass	71.03	71.03	71.50	71.50
haberman	74.51	71.90	74.18	73.86
heart-h	83.33	80.95	82.99	82.99
heart-statlog	84.82	83.70	83.33	84.82
hepatitis	82.58	83.87	81.29	85.16
ionosphere	92.59	92.31	92.02	93.16
iris	94.67	95.33	96.00	96.00
lymph	84.46	83.11	83.11	85.14
monks	99.77	99.54	99.77	99.54
postop	67.78	68.89	70.00	71.11
primary-tumor	45.72	46.31	46.90	46.02
segment	97.32	97.49	97.40	97.49
sonar	82.69	84.62	82.69	83.65
soybean	95.17	95.02	94.58	94.14
spect	84.27	83.52	83.90	85.39
tic-tac-toe	92.80	94.26	93.42	94.68
vehicle	71.63	74.82	74.47	74.94
vote	95.66	95.88	95.88	96.31
vowel	91.41	94.55	94.04	95.05
wine	97.75	98.32	97.19	97.75
wisconsin-cancer	73.43	74.48	73.08	75.52
yeast	59.77	60.31	59.97	60.78
yugoslavia-cancer	96.28	96.42	96.57	96.42
zoo	96.04	96.04	96.04	97.03
Average:	83.57	83.76	83.76	84.32

Table 7.8: Comparison of Predictive Accuracies of Aggregate Certainty Estimators with Additional Baseline Strategies

Data Set	Standard Voting	Accuracy Weighted	Select Best	Arbitration	Modified Stacking	Aggregate Certainty Estimators
anneal	99.22	99.33	98.89	99.22	99.11	99.33
audiology	78.76	79.20	78.76	80.09	75.66	78.32
balance-scale	89.28	89.60	89.92	89.44	95.52	89.92
bupa	68.99	69.28	67.25	66.38	57.10	71.01
car	96.30	96.30	98.73	96.18	99.13	97.74
cmc	53.70	52.89	52.61	56.14	50.44	53.50
colic	83.97	83.97	83.70	82.34	82.07	84.24
credit-a	85.65	85.65	84.78	85.51	84.64	86.23
credit-g	75.30	75.30	75.30	75.00	73.30	75.40
dermatology	97.81	97.81	96.18	97.00	96.72	97.27
diabetes	76.56	76.56	74.35	77.34	71.22	76.56
ecoli-c	86.31	87.20	86.61	86.31	84.82	87.50
glass	71.50	73.36	70.09	67.76	71.50	71.50
haberman	74.18	74.18	74.84	71.57	71.90	73.86
heart-h	82.99	82.99	84.35	81.29	80.27	82.99
heart-statlog	83.70	83.70	81.48	84.44	79.26	84.82
hepatitis	81.29	81.29	83.23	83.87	81.94	85.16
ionosphere	92.02	92.02	88.32	92.02	93.16	93.16
iris	95.33	96.00	90.00	96.00	95.33	96.00
lymph	81.76	81.76	77.70	82.43	83.11	85.14
monks	99.77	99.77	100.00	100.00	100.00	99.54
postop	70.00	70.00	70.00	70.00	71.11	71.11
primary-tumor	48.08	47.49	51.03	47.49	37.46	46.02
segment	97.32	97.49	96.32	96.84	97.40	97.49
sonar	82.69	82.69	83.65	81.73	86.06	83.65
soybean	94.44	94.14	92.83	94.88	93.85	94.14
spect	83.90	83.90	83.52	83.15	79.40	85.39
tic-tac-toe	93.32	93.32	98.96	96.56	99.79	94.68
vehicle	75.65	75.65	79.20	74.47	80.73	74.94
vote	95.88	95.88	96.10	96.53	97.18	96.31
vowel	93.64	94.65	96.06	94.95	96.67	95.05
wine	97.19	97.19	97.75	97.19	96.63	97.75
wisconsin-cancer	73.43	73.43	75.18	73.43	74.48	75.52
yeast	59.70	59.70	58.42	60.45	57.62	60.78
yugoslavia-cancer	96.57	96.57	97.00	95.99	97.43	96.42
zoo	95.05	95.05	92.08	95.05	93.07	97.03
Average:	83.65	83.76	83.48	83.58	82.92	84.32

7.5.2 Discussion

Aggregate Certainty Estimators is able to achieve higher average classification accuracy than any of three standard baseline strategies over the 36 data sets studied. A comparison between Table 7.7 and Table 7.8 shows that using single certainty estimators in weighting the votes of an ensemble can allow the ensemble to make improvements in average predictive accuracy. Two of the three single certainty estimator ensembles can achieve higher average classification accuracy than a baseline strategy of standard voting. However, the use of these single certainty estimator values is not sufficient to create an ensemble that can produce a higher average predictive accuracy on a level that is statistically significant, so investigation into additional certainty estimators is warranted.

The higher average accuracy of Aggregate Certainty Estimators does come with a higher cost of computation, but for two-thirds of the certainty estimators, the increase in computational complexity is only linear in regards to the size of the data set. The other one-third of the certainty estimators requires a cross validation strategy in the training set. The computational complexity for these certainty estimators could be reduced substantially by reducing the number of folds used in the calculations.

7.6 Conclusion and Future Work

This work presents a viable new method of combining the outputs of base inducers in a multi-classifier system using multiple certainty estimators to predict certainty in the classification of a given instance. A number of certainty estimators designed for this task are proposed for each of five different types of classifiers. Aggregate measures are shown to be more highly correlated with whether an instance is correctly classified than any of the individual measures. The strategy of Aggregate Certainty Estimators, which employs all of the certainty estimators presented, is shown to achieve a higher average classification accuracy over 36 data sets than five alternate ensemble strategies.

The certainty estimators presented in this work explore some of the strengths and weaknesses of a given classifier on a given data set. This information could result in the development of new algorithms. For example, a new instance-based classifier might be developed in which only instances that were correctly classified in hold-one-out cross validation would be allowed to vote on the classification of an unseen instance. The probabilities output by a Naïve Bayes classifier might be altered slightly based on information gained through certainty estimators like the ones presented here. Insights gained by observing the behavior of the certainty estimators on various data sets may help target areas of improvement to increase classification accuracy of individual classifiers.

Appendix

Tables 7.9 through 7.13 offer more rigorous presentations of how the estimators are calculated. When calculating certainty measures for training set instances to determine correlation values, “training set” in these formulas refers to the instances used for training in hold-one-out cross-validation and “test instance” refers to the instance being held out (e.g. If 135 instances of the iris data set were being used to evaluate the other 15 instances in ten-fold cross-validation, measures for the 135 training set instances would be calculated using 134 of them as a “training set” subset and the remaining instances as a “test instance”). When calculating certainty measures for the test instances, “training set” in these formulas refers to the standard cross-validation training set (e.g. The entire set of 135 iris instances) and “test instance” refers to the instances in the fold being evaluated (e.g. The remaining 15 iris instances).

Table 7.9: Certainty Estimators for Decision Tree

1. Purity of Classification: $\frac{p}{m}$

p : number of training set instances with predicted class
at leaf node where test instance is classified
m : number of training set instances at leaf node where test instance is classified

2. Instances at Leaf Node: $\frac{m}{n}$

m : number of training set instances at leaf node where test instance is classified
n : number of instances in the training set

3. Level of Leaf Node : $\frac{(k-v)}{k}$

k : maximum number of levels in the tree
v : level of leaf node where test instance is classified

4. Information Gain along Path: $\sum\limits_{1}^{v} g_i * \frac{1}{v} * \frac{1}{h}$

g_i : information gained by splitting on the selected attribute at level i in the tree
v : level of leaf node where test instance is classified
h : maximum possible information gain for given training set

Information gain for an attribute A given data set S is defined as follows:
$InformationGain(A) = Entropy(S) - \Sigma_{i=1}^{|A|} \frac{|S_i|}{|S|} Entropy(S_i)$

5. Correctly Classified Instances: $\frac{q}{m}$

q : number of training set instances at leaf node where test instance is classified
that were correctly classified in cross-validation on the training set
m : number of training set instances at leaf node where instance is classified

6. Correctly Classified Voters: $\frac{r}{p}$

r : number of training set instances with predicted class
at leaf node where test instance is classified
that were correctly classified in cross-validation on the training set
p : number of training set instances with predicted class
at leaf node where test instance is classified

Table 7.10: Certainty Estimators for Rule-Based Classifier

1. Purity of Classification: $\frac{p}{m}$

p : number of training set instances with predicted class
covered by the rule applying to the test instance
m : number of training set instances covered by the rule applying to the test instance

2. Number of Antecedents: $\frac{(k-v)}{k}$

k : maximum number of antecedents in any rule of the table
v : number of antecedents in the rule applying to the test instance

3. Number of Instances Covered: $\frac{m}{n}$

m : number of training set instances covered by the rule applying to the test instance
n : number of instances in the training set

4. Correctly Classified Instances: $\frac{q}{n}$

q : number of training set instances covered by the rule applying to the test instance
that were correctly classified in cross-validation on the training set
n : number of training set instances covered by the rule applying to the test instance

5. Correctly Classified Voters: $\frac{r}{p}$

r : number of training set instances with predicted class
covered by the rule applying to the test instance
that were correctly classified in cross-validation on the training set
p : number of training set instances with predicted class
covered by the rule applying to the test instance

6. Instance is Covered by Rule: $\begin{cases} 1.0 : \text{A rule in the table applies to the test instance} \\ 0.0 : \text{No rules in the table apply to the test instance} \end{cases}$

Table 7.11: Certainty Estimators for Instance-Based Classifier

1. Neighbors in Agreement: $\frac{p}{m}$

p : number of neighbors with predicted class
m : number of neighbors considered (5)

2. Highest Minus Second: $w_1 - w_2$

w_1 : distance-weighted vote which determines the predicted class
w_2 : distance-weighted vote of the next highest class

3. Average Distance to Neighbors: $1 - \frac{c}{d}$

c : average distance to five neighbors
d : average distance to all instances in training set

4. Correctly Classified Neighbors: $\frac{q}{m}$

q : number of neighbors that were correctly classified in cross-validation on the training set
m : number of neighbors considered (5)

5. Correctly Classified Voters: $\frac{r}{p}$

r : number of neighbors with predicted class that were correctly classified in cross-validation on the training set
p : number of neighbors with predicted class

6. 3-NN vs. 5-NN vs. 7-NN: $\begin{cases} 1.0 : \text{3-NN and 7-NN match 5-NN classification} \\ 0.5 : \text{one matches} \\ 0.0 : \text{neither matches} \end{cases}$

Table 7.12: Certainty Estimators for Naïve Bayes Classifier

1. Probability of Class Value: b_1 b_i : ith ordered value of label assigned to test instance (ranked by probability)
2. Highest Minus Second: $b_1 - b_2$ b_i : ith ordered value of label assigned to test instance (ranked by probability)
3. Highest Minus Remaining $b_1 - \sum\limits_{3}^{n} b_i$ b_i : ith ordered value of label assigned to test instance (ranked by probability)
4. Value Probability Averages $(\Sigma_{i=1}^{m}(v_i/n))/m$ v_i : number of training set instances that have the ith attribute value in common with the test set instance n : number of instances in the training set m : number of attributes in any instance
5. Correctly Classified Neighbors: $\frac{q}{m}$ q : number of neighbors that were correctly classified in cross-validation on the training set m : number of neighbors considered (5) Note: Neighbors are calculated based on similarity of predicted probability
6. Correctly Classified Voters: $\frac{r}{p}$ r : number of neighbors with predicted class that were correctly classified in cross-validation on the training set p : number of neighbors with predicted class Note: Neighbors are calculated based on similarity of predicted probability

Table 7.13: Certainty Estimators for Multilayer Perceptron

1. Activation Output: a_1

a_1 : highest activation output

2. Highest Minus Second: $a_1 - a_2$

a_1 : highest activation output
a_2 : second highest activation output

3. Correctly Classified Neighbors: $\frac{p}{m}$

p : number of neighbors that were correctly classified in cross-validation on the training set
m : number of neighbors considered (5)

Note: Neighbors are calculated based on similarity of activation outputs

4. Correctly Classified Neighbors (Hidden Layer): $\frac{p_h}{m}$

p_h : number of neighbors that were correctly classified in cross-validation on the training set
m : number of neighbors considered (5)

Note: Neighbors are calculated based on similarity of hidden layer activation outputs

5. Average Distance to Neighbors $1 - \frac{c}{d}$

c : average distance to five neighbors
d : average distance to all instances in training set

Note: Neighbors are based on similarity of output layer activation

6. Average Distance to Neighbors (Hidden Layer) $1 - \frac{c_h}{d_h}$

c_h : average distance to five neighbors
d_h : average distance to all instances in training set

Note: Neighbors are based on similarity of hidden layer activation outputs

Chapter 8

Bayesian Model Combination

"Imagine if every Thursday your shoes exploded if you tied them the usual way. This happens to us all the time with computers, and nobody thinks of complaining." **–Jef Raskin**

K. Monteith, J. Carroll, N. Toronto, K. Seppi, T. Martinez. The Problem with Bayesian Model Averaging (And How to Fix It). *In submission.*

Abstract: *Bayesian methods are theoretically optimal for achieving high learner accuracy. Bayesian model averaging is generally considered the standard model for creating ensembles of learners using Bayesian methods, but this technique is often out-performed by more ad hoc methods in empirical studies. It has been proposed that Bayesian model averaging struggles in practice because it accounts for uncertainty about which model is correct but still operates under the assumption that only one of them is. This work provides empirical verification for this hypothesis using several different Bayesian model combination approaches tested on a wide variety of classification problems. The results suggest that, in order to more effectively access the benefits inherent in ensembles, Bayesian strategies should be directed more towards model combination rather than the model selection implicit in Bayesian model averaging. We show that even the most simplistic of Bayesian model combination strategies can compete with the traditional ad hoc techniques of bagging and boosting, as well as significantly outperforming BMA over a wide variety of cases.*

8.1 Introduction

Learner error can often be reduced by combining information from a set of models. This poses the challenge of finding effective ways to create combinations of learners. A number of *ad hoc* strategies have been proposed to address this task. For example, bagging [Breiman, 1996] employs one of the simplest methods of combining the information presented in an ensemble: allowing each learner to have one vote toward the final classification of an instance. Boosting [Freund and Schapire, 1996], attempts to focus on harder instances during the course of training, and votes are weighted by the accuracy that a given learner achieves on the data set.

One possible explanation for the success of ensemble learners is based on Bayesian learning theory [Bernardo and Smith, 1994]. Using a single model for learning ignores the uncertainty about model correctness that results from a finite amount of data. Strategies such as bagging compensate for this uncertainty simply by incorporating a set of models into the learning process. Bayesian model averaging (BMA), the generally accepted method for applying Bayesian learning theory to the task of model combination, accounts for uncertainty of model correctness by integrating over the model space and weighting each model by the probability of its being the "correct" model.

One might expect Bayesian model averaging to perform well since Bayesian techniques have been applied to many other tasks with high success. For example, even simple single model classifiers such as Naïve Bayes [Lang, 1995] and Flexible Bayes [John and Langley, 1995] can achieve remarkably high accuracy on certain problems. More complex distributions can be represented by Bayesian mixture models. Techniques such as Markov Chain Monte Carlo can be used to identify parameters for subdistributions in an overall model [Gilks, 2005]. Specific models are also commonly used for specific tasks. The latent Dirichlet allocation model is commonly used to identify topics present in a set of documents [Blei et al., 2003]. However, when it comes to the task of ensemble creation, the standard technique of Bayesian model averaging encounters some problems.

In an empirical study, Domingos [2000] showed that Bayesian model averaging is prone to higher error rates than more *ad hoc* methods. Specifically, Bayesian model averaging resulted in higher average error rates than bagging and partitioning in a variety of experiments. These results are surprising given the supposed optimality of Bayesian techniques and their success in so many other areas.

Domingos argued that the problem with BMA is that it places too much weight on the maximum likelihood classifier. Even slight differences in error rate between classifiers result in much higher weighting of the more accurate classifier in the ensemble. Yet Bayesian model averaging is theoretically the optimal method for dealing with uncertainty about which hypothesis in the hypothesis space is correct. To account for the superior performance of *ad hoc* methods in empirical studies, we must turn to an alternate explanation for the success of ensembles.

Minka [December 2000] theorized that Bayesian model averaging is outperformed by other strategies because it fails to take advantage of the enriched hypothesis space of the ensemble. If Minka is correct, an ensemble does more than just deal with uncertainty about which model is the correct model; it can augment the hypothesis space with hypotheses that its individual members may not be able to represent on their own. Further, ensembles may change the preferential bias of a learning algorithm, predisposing the algorithm towards combinations of models that tend to overfit less than single learners. As Minka states in his paper, "...the only flaw with BMA is the belief that it is an algorithm for model combination." Bagging and other *ad hoc* strategies may have an advantage over Bayesian model averaging because they incorporate more information from the enriched hypothesis space of an ensemble. This suggests that if Bayesian methods are to be effectively used in ensemble creation strategies, efforts should be directed towards creation of Bayesian mixture models that learn the optimal combination of individual members of the ensemble. Such strategies would take advantage of both the optimality of Bayesian learning strategies and the error reduction advantages that can result from combinations of models.

This paper provides empirical evidence for Minka's hypothesis. In Section 8.2 we review Domingos' argument that Bayesian model averaging assumes a single ensemble member to be correct. Sections 8.3 and 8.4 further investigate the behavior of Bayesian model averaging under different conditions. Section 8.6 then proposes several possibilities for models that employ the same principles as Bayesian model averaging, but direct them towards the task of model combination instead of model selection. More complicated strategies are clearly possible, but even the simple models presented here significantly outperforms Bayesian model averaging in terms of error reduction on 50 data sets. As a complement to these techniques, we present a strategy in Section 8.7 that uses Bayesian methods to learn optimal component models given a fixed combination of weights. Again, while there is clear potential for more sophisticated strategies, even this simple one outperforms more *ad hoc* methods of model learning in terms of error reduction.

8.2 The Problem with Bayesian Model Averaging

With traditional Bayesian model averaging, the class value assigned to a given example by the overall model is determined by taking the probability of each class value as predicted by a single model, multiplying by the probability of that model given the data, and summing these values for all models in the hypothesis space. Let n be the size of a data set D. Each individual example d_i is comprised of a vector of attribute values x_i and an associated class value y_i. The model space is approximated by a finite set of learners, H, with h being an individual hypothesis in that space. Equation 8.1 illustrates how the probability of a class value is determined for a given example. The class value assigned to the instance will be the one with the maximum probability.

$$p(y_i|x_i, D, H) = \sum_{h \in H} p(y_i|x_i, h)p(h|D) \tag{8.1}$$

By Bayes' Theorem, the *posterior probability* of h given D can be calculated as shown in Equation 8.2. Here, $p(h)$ represents the *prior probability* of h and the product of the $p(d_i|h)$ determines the *likelihood*.

$$p(h|D) = \frac{p(h)}{p(D)} \prod_{i=1}^{n} p(d_i|h) \tag{8.2}$$

Bayesian model averaging strategies commonly assume a *uniform class noise model* when determining likelihood [Domingos, 2000]. With this model, the class of each example is assumed to be corrupted with probability ϵ. This means that $p(d_i|h)$ is $1-\epsilon$ if h correctly predicts class y_i for example x_i and ϵ otherwise. Equation 8.2 can be rewritten as shown in Equation 8.3. (Since the prior probability of the data $p(D)$ is the same for each model, the equation becomes a statement of proportionality and $p(D)$ can be ignored.)

$$p(h|D) \propto p(h)(1-\epsilon)^r(\epsilon)^{n-r} \tag{8.3}$$

r is the number of examples correctly classified by h. ϵ can be estimated by the average error rate of the model on the data. This method of calculating likelihood tends to weight even slightly more accurate classifiers much more heavily. For example, on a data set with 100 examples, a learner that achieved 95% accuracy would be weighted as 17 times more likely than a learner that achieved an accuracy of 94%.

$$(1-\tfrac{5}{100})^{95}(\tfrac{5}{100})^5 = 2.39 * 10^{-9}$$

$$(1-\tfrac{6}{100})^{94}(\tfrac{6}{100})^6 = 1.39 * 10^{-10}$$

Using these posterior probabilities to weight learner classifications is clearly an effective way of exploiting the model with the highest accuracy while still allowing influence from other models to account for the uncertainty about which model is correct. It is somewhat ineffective, however, at taking advantage of information provided by the entire set of models. In practice, this inefficiency results in lower performance in terms of average accuracy on various classification tasks.

Experiments were conducted on the twenty-six data sets cited by Domingos, but since this selection of data sets proved insufficient to draw conclusions about the statistical significance of mean differences in accuracy, an additional twenty-four datasets were included. All data sets were obtained from the UCI repository [Hettich et al., 1998]. Error was calculated using ten-fold cross-validation. Information about these data sets is provided in Table 8.1.

Table 8.2 compares BMA to the strategies of bagging, boosting, and stacking as well as to a single decision-tree classifier. Experiments were implemented using modified Weka code [Witten and Frank, 2005]. Ten J48 decision trees (Weka's implementation of the C4.5 algorithm) with reduced-error pruning were used as the base classifiers in each of the algorithms. Bagging and boosting were implemented using Weka defaults. For bagging, training data for the component classifiers was obtained by drawing with replacement from the initial training set until a new training set the same size as the original set was created [Breiman, 1996]. Training sets for the boosting algorithm were generated in a similar manner, but instances misclassified by initial component classifiers were more likely to appear in the training data for subsequent classifiers [Freund and Schapire, 1996]. Stacking and BMA were implemented using the same ten decision trees that were used in the bagging experiments as component classifiers. Stacking was implemented by treating the probability distribution outputs of these classifiers as inputs to a decision-tree meta-classifier. Probabilities for class predictions by individual learners and ensembles were estimated using Weka defaults. For the individual J48 decision trees, $p(y_i|x_i, h)$ was estimated based on the purity of classification at the leaf node.

Just as in Domingo's experiments, these results show that Bayesian model averaging achieves a lower average accuracy on the data sets than bagging or even a single classifier. It also achieves lower average accuracy than boosting and stacking.

Table 8.1: Information for Data Sets

	Number of Instances	Number of Attributes	Number of Classes	Attribute Type	Missing Attributes?
anneal	898	39	6	mixed	no
audiology	226	70	24	discrete	yes
autos	205	26	7	mixed	no
balance-scale	625	5	3	real	no
bupa	345	7	2	real	no
cancer1	286	10	2	discrete	yes
cancer2	699	10	2	real	no
car	1728	7	4	discrete	no
cmc	1473	10	3	mixed	no
credit-a	692	16	2	mixed	yes
credit-g	1000	21	2	mixed	no
dermatology	366	35	6	mixed	yes
diabetes	768	9	2	real	no
echo	132	13	3	mixed	yes
ecoli-c	336	8	8	real	no
glass	214	10	7	real	no
haberman	306	4	2	real	no
heart-cleveland	303	14	5	mixed	yes
heart-h	294	14	5	mixed	yes
heart-statlog	270	14	2	real	no
hepatitis	155	20	2	mixed	yes
horse-colic	368	23	2	mixed	yes
hypothyroid	3772	30	4	mixed	yes
ionosphere	352	35	2	real	no
iris	150	5	3	real	no
kr-vs-kp	3196	37	2	discrete	no
labor	57	17	2	mixed	yes
led	1000	8	10	discrete	no
lenses	24	5	3	discrete	no
liver-disorders	345	7	2	real	no
lungcancer	32	57	3	discrete	no
lymph	148	19	4	mixed	no
monks	432	7	2	discrete	no
mushroom	8124	23	2	discrete	yes
page-blocks	5473	11	5	real	no
postop	90	9	3	discrete	no
primary-tumor	339	18	22	discrete	yes
promoters	106	58	2	discrete	no
segment	2310	20	7	real	no
sick	3772	30	2	mixed	yes
solar-flare	323	13	2	discrete	no
sonar	208	61	2	real	no
soybean	684	36	20	discrete	yes
spect	267	23	2	discrete	no
tic-tac-toe	958	10	2	discrete	no
vehicle	846	19	4	real	no
vote2	461	17	2	discrete	no
wine	178	14	3	real	no
yeast	1484	9	10	real	no
zoo2	101	17	7	discrete	no

Table 8.2: Average accuracy of various ensemble combination strategies

	J48	Bagging	Boosting	Stacking	BMA
anneal	98.44	98.89	99.55	99.33	99.44
audiology	77.88	79.65	84.96	81.42	80.97
autos	81.46	83.90	83.90	80.00	80.49
balance-scale	76.64	82.24	78.88	80.96	78.88
bupa	68.70	72.75	71.59	66.09	63.48
cancer-wisconsin	75.52	73.43	69.58	69.93	68.18
cancer-yugoslavia	93.85	95.85	95.71	93.85	93.42
car	92.36	93.52	96.12	94.91	92.19
cmc	52.14	54.11	50.78	49.76	54.11
credit-a	86.09	85.36	84.20	84.35	84.35
credit-g	70.50	74.00	69.60	69.90	67.70
dermatology	93.99	95.08	95.63	94.81	93.44
diabetes	73.83	74.09	72.40	70.05	69.92
echo	97.30	95.95	95.95	97.30	97.30
ecoli-c	84.23	84.82	81.25	83.04	81.85
glass	66.82	71.03	74.30	65.89	64.95
haberman	71.90	74.84	72.55	68.30	71.24
heart-cleveland	77.56	79.21	82.18	77.23	74.26
heart-h	80.95	78.91	78.57	79.25	78.23
heart-statlog	76.67	80.00	80.37	77.41	75.19
hepatitis	83.87	83.23	85.81	79.35	81.29
horse-colic	85.33	85.60	83.42	83.15	83.15
hypothyroid	99.58	99.58	99.58	99.58	99.63
ionosphere	91.45	93.16	93.16	90.60	90.88
iris	96.00	95.33	93.33	93.33	94.00
kr-vs-kp	99.44	99.44	99.50	99.44	99.44
labor	73.68	84.21	89.47	77.19	82.46
led	100.00	100.00	100.00	100.00	100.00
lenses	83.33	79.17	70.83	70.83	79.17
liver-disorders	68.70	72.75	71.59	66.09	63.48
lungcancer	50.00	56.25	53.13	53.13	46.88
lymph	77.03	79.05	81.08	81.76	77.70
monks	96.53	100.00	100.00	100.00	100.00
mushroom	100.00	100.00	100.00	100.00	100.00
page-blocks	96.88	97.26	97.02	96.78	96.80
postop	70.00	68.89	56.67	62.22	58.89
primary-tumor	39.82	42.18	40.12	37.17	36.87
promoters	81.13	80.19	85.85	78.30	79.25
segment	96.93	97.40	98.48	97.01	96.36
sick	98.81	98.73	99.18	99.05	99.02
solar-flare	97.83	97.83	96.59	97.83	97.83
sonar	71.15	74.52	77.88	74.52	73.08
soybean	91.51	93.27	92.83	92.39	90.19
spect	78.28	80.52	80.15	77.53	77.90
tic-tac-toe	85.07	92.80	96.35	93.11	86.22
vehicle	72.46	76.60	76.24	72.81	73.40
vote	94.79	95.23	95.66	95.01	94.79
wine	93.82	94.94	96.63	94.38	93.82
yeast	56.00	60.78	56.40	54.58	55.73
zoo	92.08	94.06	96.04	91.09	90.10
Average:	82.37	84.01	83.62	81.84	81.36

8.3 What about a Different Likelihood Function?

Since the *uniform class noise model* places so much emphasis on the maximum likelihood option, one possible approach to modifying BMA might be using a different method of calculating likelihood. Many classifiers output probability distributions over possible class values, and classifiers that do not output true probabilities can often generate decent approximations. For example, approximate probability distributions for decision tree classification can be obtained by calculating the percentages of class values appearing in a given lead node of the tree. If probability distributions from base classifiers are available, the calculation in equation 8.3 can be replaced by the following:

$$p(h|D) \propto p(h) \prod_{i=1}^{n} y_i \tag{8.4}$$

where y_i is the probability output by h for the actual class label of x_i.

In practice, many of the probability distributions output by the decision trees used in these experiments were too stark, assigning all the probability to one class. With so many zero probability values getting included in the product for calculating likelihood, it was not uncommon for the final likelihood value to be zero for some, or even all, of the base classifiers. Table 8.3 shows the results of this new method of calculating likelihood if all of the base classifiers are assigned a default minimum likelihood value whenever this value is calculated to be zero. The results look promising–BMA now appears competitive with bagging. However, closer inspection reveals that, in this case, most of the likelihood values assigned to the base classifiers are the default minimum values. In fact, for 20 of the 50 datasets, all of the base classifiers were weighted equally because the likelihood values were all initially calculated to be zero. BMA is now competitive with bagging because it is, in essence, using the same strategy of giving equal weight to all the base classifiers.

The third column in table 8.3 shows the results when the probability distributions output by the base classifiers are “backed off” by adding one sample from each class into each leaf node's estimation of probability. This eliminates the zero-probability problem for base classifier weights. However, BMA once again struggles with the problem of assigning too much weight to the maximum likelihood classifier in the ensemble.

8.4 What If Model Selection is the Right Thing to Do?

Table 8.4 shows the classification accuracy on data sets that have been modified such that one of the base classifiers perfectly classifies both training and test set data. With one perfectly accurate classifier in an ensemble, it is no surprise that average accuracy levels for bagging and stacking on these relabeled data sets are significantly higher than those achieved on the original data. With BMA, however, one perfectly accurate base classifier allows the overall ensemble to achieve 100% accuracy on nearly all the new data sets.

8.5 Using Combinations as Base Classifiers

Since BMA is, in essence, selecting only one of the base classifiers, a better attempt at improving classification accuracy of BMA might be to use base classifiers that are already employing a combination strategy. Table 8.5 reports the results of using a set of ten bagged decision trees as each of the ten base classifiers and then combining their outputs using Bayesian Model Averaging. Now that BMA is relegating most of the classifying power to a group of classifiers instead of an individual classifier, average classification accuracy on the fifty data sets is considerably higher. Not surprisingly, average accuracy of BMA with bagged decision trees as base classifiers was significantly greater than that of BMA using single decision trees as base classifiers. In fact, the average accuracy of BMA with bagged decision trees was not significantly less than the average accuracy achieved by the strategies of bagging or boosting on the fifty datasets studied. When BMA is allowed to select between model combinations, it starts to be competitive with some of the most successful *ad hoc* strategies of model combination.

Table 8.3: Average accuracy of BMA using various methods of calculating likelihood

	BMA	BMA	BMA
	Uniform class	New likelihood	New likelihood
	noise model	calculation	calculation (adjusted)
anneal	99.44	98.22	99.33
audiology	80.97	79.65	79.65
autos	80.49	82.93	78.54
balance-scale	78.88	82.08	79.04
bupa	63.48	71.88	63.48
cancer-wisconsin	68.18	73.08	70.28
cancer-yugoslavia	93.42	94.99	93.71
car	92.19	92.88	91.96
cmc	54.11	54.18	48.81
credit-a	84.35	84.78	84.64
credit-g	67.70	73.80	67.20
dermatology	93.44	93.99	93.17
diabetes	69.92	74.35	70.31
echo	97.30	97.30	97.30
ecoli-c	81.85	85.12	81.85
glass	64.95	71.50	64.95
haberman	71.24	69.93	69.93
heart-cleveland	74.26	80.20	74.26
heart-h	78.23	77.89	76.87
heart-statlog	75.19	79.26	73.70
hepatitis	81.29	84.52	83.87
horse-colic	83.15	83.42	83.15
hypothyroid	99.63	99.60	99.55
ionosphere	90.88	92.02	89.46
iris	94.00	94.00	94.00
kr-vs-kp	99.44	99.34	99.41
labor	82.46	75.44	78.95
led	100.00	100.00	100.00
lenses	79.17	75.00	79.17
liver-disorders	63.48	71.88	63.48
lungcancer	46.88	53.13	43.75
lymph	77.70	78.38	79.05
monks	100.00	100.00	100.00
mushroom	100.00	100.00	100.00
page-blocks	96.80	97.24	96.86
postop	58.89	68.89	55.56
primary-tumor	36.87	41.89	35.69
promoters	79.25	76.42	77.36
segment	96.36	97.58	96.54
sick	99.02	98.49	98.83
solar-flare	97.83	97.83	97.21
sonar	73.08	75.00	69.71
soybean	90.19	92.24	90.19
spect	77.90	78.28	79.03
tic-tac-toe	86.22	91.02	86.53
vehicle	73.40	77.54	74.35
vote	94.79	93.93	94.36
wine	93.82	92.13	92.70
yeast	55.73	60.24	55.12
zoo	90.10	95.05	92.08
Average:	81.36	83.17	80.90

Table 8.4: Average accuracy when one of the base classifiers perfectly classifies data

	Bagging	Stacking	BMA
anneal	99.33	98.89	99.78
audiology	88.05	83.19	100.00
autos	82.44	77.56	100.00
balance-scale	89.28	83.04	100.00
bupa	79.13	68.99	100.00
cancer-wisconsin	91.96	85.66	100.00
cancer-yugoslavia	96.57	96.28	100.00
car	94.27	93.75	100.00
cmc	71.62	65.44	100.00
credit-a	92.90	90.14	100.00
credit-g	84.40	78.30	100.00
dermatology	98.36	95.36	99.45
diabetes	82.68	80.21	100.00
echo	96.21	96.97	96.97
ecoli-c	89.29	85.12	100.00
glass	81.78	73.83	100.00
haberman	88.56	85.95	100.00
heart-cleveland	85.15	81.19	100.00
heart-h	88.10	85.71	100.00
heart-statlog	85.93	84.81	100.00
hepatitis	85.81	83.23	100.00
horse-colic	96.20	93.75	100.00
hypothyroid	99.81	99.92	100.00
ionosphere	94.59	90.88	100.00
iris	98.00	96.00	100.00
kr-vs-kp	99.84	99.59	100.00
labor	94.74	84.21	98.25
led	100.00	100.00	100.00
lenses	83.33	91.67	100.00
liver-disorders	79.13	68.99	100.00
lungcancer	62.50	56.25	100.00
lymph	83.11	79.05	100.00
monks	95.60	95.60	95.60
mushroom	100.00	100.00	100.00
page-blocks	98.70	98.23	100.00
postop	92.22	84.44	100.00
primary-tumor	68.44	57.52	100.00
promoters	85.85	78.30	99.06
segment	96.58	96.06	100.00
sick	99.18	99.13	100.00
solar-flare	100.00	100.00	100.00
sonar	80.77	72.12	100.00
soybean	94.29	91.95	100.00
spect	93.63	88.39	100.00
tic-tac-toe	88.41	87.27	100.00
vehicle	82.86	81.09	100.00
vote	98.26	98.48	100.00
wine	92.13	91.01	100.00
yeast	73.99	64.96	100.00
zoo	95.05	96.04	100.00
Average:	89.58	86.29	99.78

Table 8.5: Average accuracy of BMA using bagged decision trees as base classifiers

	BMA	BMA
	Base classifier:	Base classifier:
	Single decision tree	Ten bagged decision trees
anneal	99.44	99.00
audiology	80.97	80.97
autos	80.49	76.10
balance-scale	78.88	81.44
bupa	63.48	66.38
cancer-wisconsin	68.18	71.33
cancer-yugoslavia	93.42	95.71
car	92.19	93.69
cmc	54.11	52.61
credit-a	84.35	84.93
credit-g	67.7	72.50
dermatology	93.44	96.72
diabetes	69.92	73.57
echo	97.30	98.65
ecoli-c	81.85	84.52
glass	64.95	67.76
haberman	71.24	70.26
heart-cleveland	74.26	79.87
heart-h	78.23	79.25
heart-statlog	75.19	80.00
hepatitis	81.29	80.65
horse-colic	83.15	84.51
hypothyroid	99.63	99.50
ionosphere	90.88	92.59
iris	94.00	94.00
kr-vs-kp	99.44	99.50
labor	82.46	84.21
led	100.00	100.00
lenses	79.17	79.17
liver-disorders	63.48	66.38
lungcancer	46.88	53.13
lymph	77.70	80.41
monks	100.00	100.00
mushroom	100.00	100.00
page-blocks	96.80	97.41
postop	58.89	63.33
primary-tumor	36.87	43.66
promoters	79.25	80.19
segment	96.36	96.97
sick	99.02	98.75
solar-flare	97.83	97.83
sonar	73.08	78.85
soybean	90.19	91.80
spect	77.90	82.02
tic-tac-toe	86.22	93.11
vehicle	73.40	74.23
vote	94.79	95.01
wine	93.82	94.38
yeast	55.73	56.74
zoo	90.10	95.05
Average:	81.36	83.17

8.6 Bayesian Model Combination

The results in the previous section were obtained using a total of one hundred decision trees in the base classifiers. Higher classification accuracy can be obtained using just the initial ten base classifiers if BMA is modified to select from *combinations* of these ten base classifiers instead of the classifiers themselves. This strategy is referred to here as Bayesian model combination (BMC). Equation 8.1 is modified as follows:

$$p(y_i|x_i, D, H, E) = \sum_{e \in E} p(y_i|x_i, H, e)p(e|D) \tag{8.5}$$

where e is an element in the space E of possible model combinations. In this case, the outputs from individual hypotheses are combined in a variety of ways to create a set of diverse ensembles. The output from each ensemble is then weighted by the probability that the ensemble is correct given the training data. Now, instead of integrating out uncertainty about which ensemble *member* is correct, we are instead integrating out uncertainty about which *model combination* is correct. Figure 8.1 illustrates the process of Bayesian model averaging, and Figure 8.2 illustrates the differences with this new strategy of Bayesian model combination.

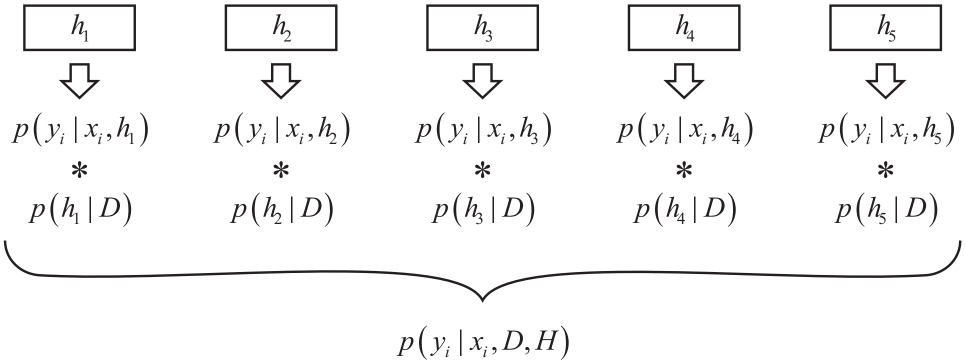


Figure 8.1: Bayesian Model Averaging. Since the probability of the most likely hypothesis is often much higher than the probability of the other hypotheses, $p(y_i|x_i, D, H)$ will be predominantly determined by $p(h_{mostLikely}|D)$.

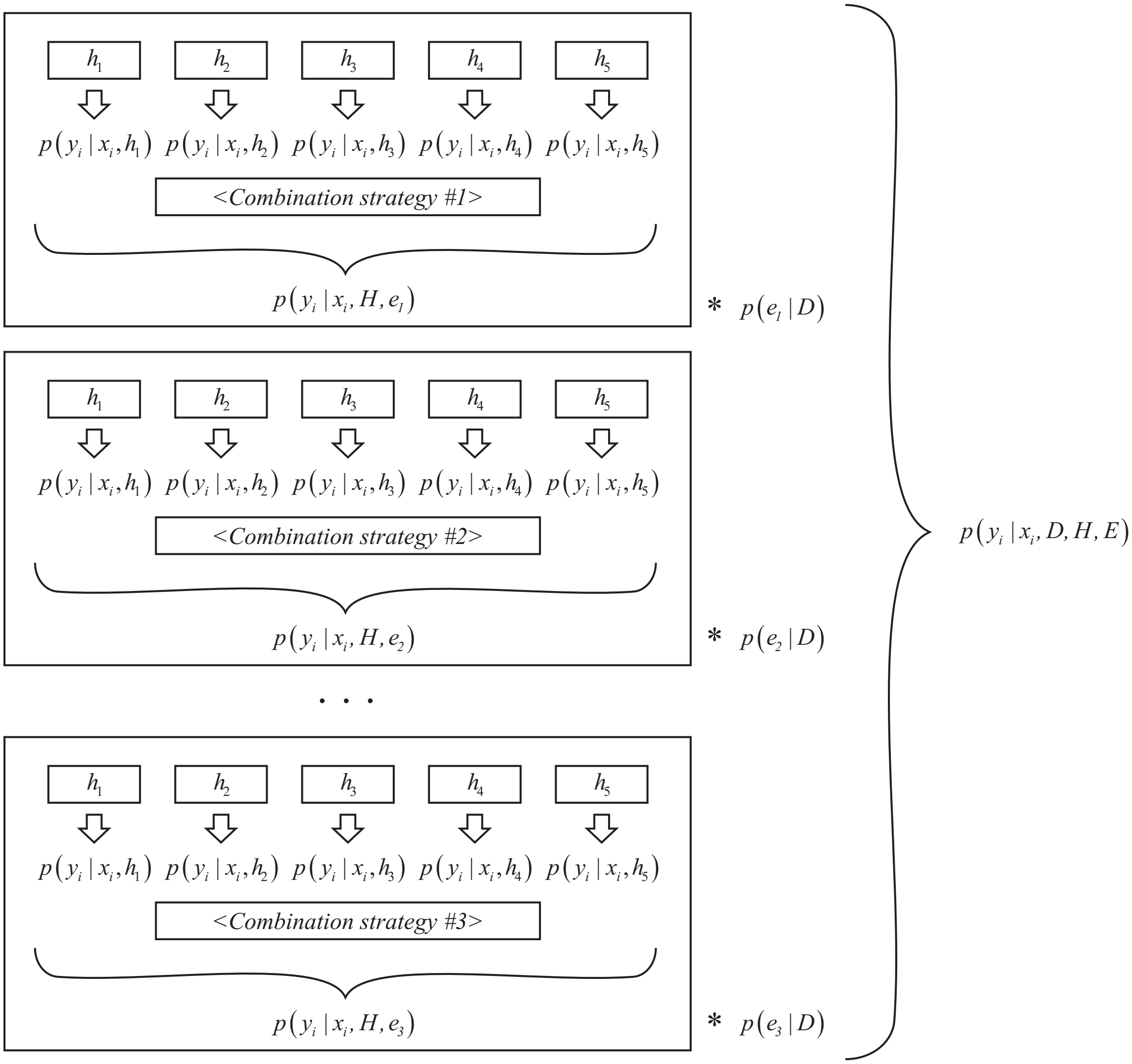


Figure 8.2: Bayesian Model Combination. In this case, $p(y_i|x_i, D, H, E)$ will be predominantly determined by $p(e_{mostLikely}|D)$. The model is now heavily weighting the most probable combination of hypotheses instead of the most probable single hypothesis.

Table 8.6: Weight assignments for individual components in a simple Bayesian model combination learner. Each component is weighted with a uniform prior in these experiments.

Raw weights	Normalized weights	$p(e)$
1 1 1 1 1 1 1 1 1 1	0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10	$\frac{1}{59049}$
1 1 1 1 1 1 1 1 1 2	0.09 0.09 0.09 0.09 0.09 0.09 0.09 0.09 0.09 0.18	$\frac{1}{59049}$
1 1 1 1 1 1 1 1 1 3	0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.25	$\frac{1}{59049}$
1 1 1 1 1 1 1 1 2 1	0.09 0.09 0.09 0.09 0.09 0.09 0.09 0.09 0.18 0.09	$\frac{1}{59049}$
. . .	. . .	. . .
3 3 3 3 3 3 3 3 3 3	0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10	$\frac{1}{59049}$

8.6.1 BMC with a Linear Combinations of Models

For the first set of Bayesian model combination experiments, ensembles were created using linear combinations of outputs from the base classifiers. Ensembles consisted of m decision trees whose votes were combined using various weights. In order to generate a diverse collection of ensembles, nested for loops were used to assign incrementally increasing values to the base components. These values were then normalized to produce a vector of weights. Table 8.6 illustrates how weights were assigned. For the reported experiments $m = 10$ and ensemble weightings were assigned using an increment value of three. This allowed for the creation of $59,049$ different ensembles from the same ten base classifiers.

Posterior probabilities for ensembles in the Bayesian model combination approach were estimated the same way they were estimated for individual learners in Bayesian model averaging. Equation 8.3 can be easily applied to calculate $p(e|D)$ instead of $p(h|D)$. The class of each example is assumed to be corrupted with probability ϵ, so $p(d_i|e)$ is $1 - \epsilon$ if e correctly predicts class y_i for example x_i and ϵ otherwise. $p(y_i|x_i, e)$ was calculated by averaging probability estimates from the individual trees.

As shown in Figure 8.7, this strategy of iterating over combinations of model significantly outperforms Bayesian model averaging.

Table 8.7: Average accuracy of BMA and BMC

	Bayesian Model Averaging	Bayesian Model Combination
anneal	99.44	98.89
audiology	80.97	82.30
autos	80.49	84.39
balance-scale	78.88	81.44
bupa	63.48	69.86
cancer-wisconsin	68.18	73.08
cancer-yugoslavia	93.42	95.42
car	92.19	93.81
cmc	54.11	53.22
credit-a	84.35	85.65
credit-g	67.70	72.90
dermatology	93.44	95.36
diabetes	69.92	72.92
echo	97.30	97.30
ecoli-c	81.85	84.82
glass	64.95	70.56
haberman	71.24	74.51
heart-cleveland	74.26	80.86
heart-h	78.23	79.59
heart-statlog	75.19	80.00
hepatitis	81.29	84.52
horse-colic	83.15	85.87
hypothyroid	99.63	99.60
ionosphere	90.88	93.16
iris	94.00	95.33
kr-vs-kp	99.44	99.44
labor	82.46	84.21
led	100.00	100.00
lenses	79.17	79.17
liver-disorders	63.48	69.86
lungcancer	46.88	53.13
lymph	77.70	79.73
monks	100.00	100.00
mushroom	100.00	100.00
page-blocks	96.80	97.26
postop	58.89	68.89
primary-tumor	36.87	41.59
promoters	79.25	81.13
segment	96.36	97.66
sick	99.02	98.94
solar-flare	97.83	97.83
sonar	73.08	75.48
soybean	90.19	93.56
spect	77.90	79.03
tic-tac-toe	86.22	93.63
vehicle	73.40	76.36
vote	94.79	95.66
wine	93.82	95.51
yeast	55.73	60.24
zoo	90.10	93.07
Average:	81.36	83.93

8.6.2 Comparison of Run Times and Accuracy Levels

It could be argued BMC is being given an unfair advantage over BMA in these experiments: despite an equal number of base classifiers, BMC is still making use of many more classifier combinations and using more computing resources. However, even if these concerns are addressed and computing time is allocated more equitably, BMC still outperforms BMA in terms of average accuracy. Figures 8.3 reports accuracy levels if BMC only considers between 10 and 100 combinations of the ten base classifier decision trees. This is compared to a version of BMA that used between 10 and 100 decision trees as base classifiers. Figure 8.4 reports the time taken to train such models. BMC achieves higher accuracy levels with lower training times. For example, BMC with 100 combinations of ten base classifiers significantly outperforms BMA with one hundred base classifiers, yet it requires a fraction of the training time.

8.6.3 BMC with Sampling from a Dirichlet Distribution

Further improvements in accuracy can be achieved by a slightly more sophisticated strategy for creating the various model combinations. Instead of assigning weights incrementally, the weights for each combination of the base classifiers can be obtained by sampling from a Dirichlet distribution.

In this next set of experiments, weights for the first q combinations were drawn from a Dirichlet distribution with uniform alpha values. $p(e|D)$ was then calculated for each combination, and the weights from the most probable combination were used to update the alpha values for the distribution from which the next q weight assignments were drawn. Table 8.8 illustrates how weights were assigned in these experiments.

The same ten base classifiers from the previous section were used in these experiments. Alpha values were updated with a q value of three, and $59,049$ Dirichlet-generated weight assignments were considered. As shown in Figure 8.9, this strategy of iterating over combinations of models allows a Bayesian method to compete with the *ad hoc* methods.

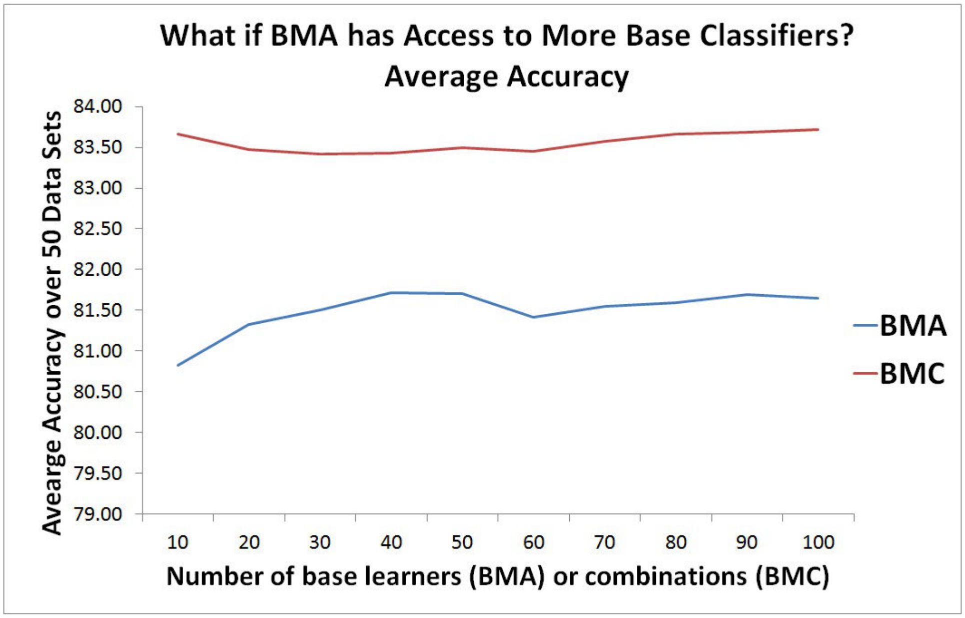


Figure 8.3: Average accuracy of BMA and BMC with varying numbers of base classifiers or classifier combinations

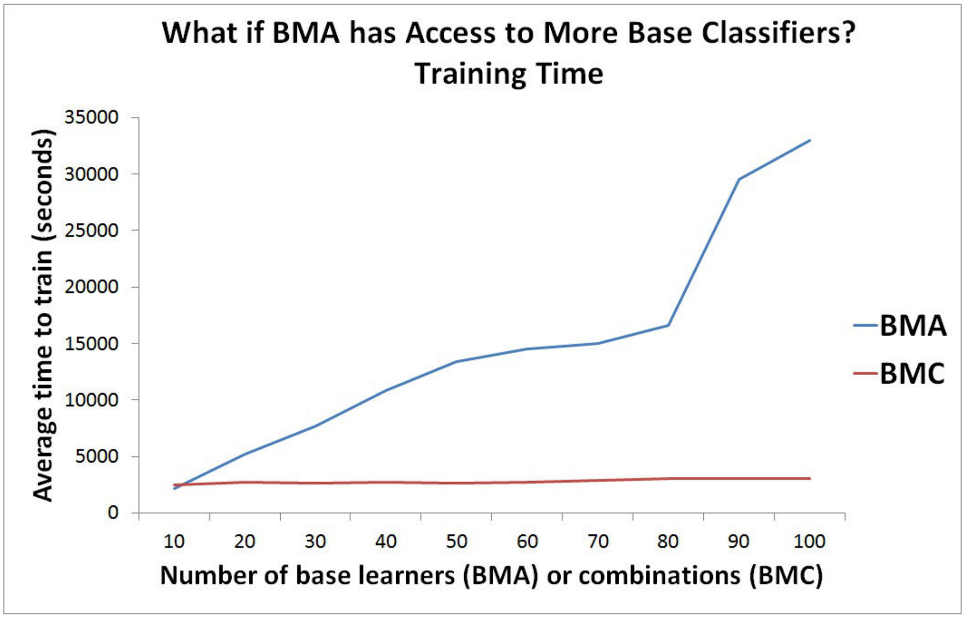


Figure 8.4: Average training time for BMA and BMC with varying numbers of base classifiers or classifier combinations

Table 8.8: Sample weight assignments for individual components in a Bayesian model combination learner employing a Dirichlet distribution. After a set of combinations are generated, the weights of the most probable combination are used to update the alpha values of the Dirichlet from which the next set of combinations will be drawn. As with the first experiments, each component is weighted with a uniform prior.

Weights	$p(e\|D)$	$p(e)$
Initial alpha values: 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00		
0.06 0.26 0.08 0.11 0.09 0.20 0.17 0.00 0.02 0.01	0.00	$\frac{1}{59049}$
0.10 0.15 0.14 0.28 0.04 0.00 0.17 0.03 0.07 0.02	0.03	$\frac{1}{59049}$
0.00 0.10 0.04 0.04 0.03 0.03 0.09 0.02 0.29 0.36	0.02	$\frac{1}{59049}$
New alpha values: 1.10 1.15 1.14 1.28 1.04 1.00 1.17 1.03 1.07 1.02		
0.07 0.00 0.04 0.12 0.26 0.15 0.07 0.13 0.01 0.13	0.03	$\frac{1}{59049}$
0.16 0.13 0.15 0.05 0.00 0.04 0.07 0.14 0.13 0.12	0.02	$\frac{1}{59049}$
0.01 0.05 0.07 0.15 0.04 0.08 0.26 0.01 0.26 0.08	0.02	$\frac{1}{59049}$
New alpha values: 1.17 1.15 1.19 1.40 1.31 1.16 1.24 1.17 1.07 1.15		
0.02 0.02 0.03 0.28 0.20 0.04 0.04 0.00 0.18 0.19	0.02	$\frac{1}{59049}$
0.35 0.12 0.13 0.06 0.08 0.07 0.09 0.02 0.06 0.01	0.00	$\frac{1}{59049}$
0.07 0.14 0.02 0.01 0.17 0.01 0.17 0.15 0.14 0.12	0.03	$\frac{1}{59049}$

An application of the Friedman test reveals significant differences in average accuracy among the various strategies. ($68.40 \sim \chi^2, DF = 5, p <= 0.01$). The Bonferroni-Dunn *post hoc* test indicates that the improvement in accuracy of Bayesian Model Combination with Dirichlet sampling exceeds the critical difference for significance at a confidence level of 95% for three of the other five non-BMC strategies (Critical difference = 0.96, Mean rank differences: 1.57, 0.17, 0.44, 1.86, 2.32).

8.6.4 Comparison with Different Types of Base Learners

The ten decision trees used in the previous experiments are all fairly accurate learners in their own right, so bagging's strategy of allowing each learner an equal vote is quite effective. Our strategy of Bayesian Model Combination provides a further advantage in a situation where the learners are not so equally balanced. Figures refBaggingDumb and 8.5 illustrate the performance of the various algorithms when weak learners are used as base classifiers. In these experiments, a number of the decision tree base classifiers are replaced by classifiers

Table 8.9: Average accuracy of various ensemble combination strategies

	J48	Bagging	Boosting	Stacking	BMA	BMC
						Sampling
anneal	98.44	98.89	99.55	99.33	99.44	98.89
audiology	77.88	79.65	84.96	81.42	80.97	82.30
autos	81.46	83.90	83.90	80.00	80.49	84.88
balance-scale	76.64	82.24	78.88	80.96	78.88	81.92
bupa	68.70	72.75	71.59	66.09	63.48	71.88
cancer-wisconsin	75.52	73.43	69.58	69.93	68.18	73.08
cancer-yugoslavia	93.85	95.85	95.71	93.85	93.42	95.14
car	92.36	93.52	96.12	94.91	92.19	93.75
cmc	52.14	54.11	50.78	49.76	54.11	52.95
credit-a	86.09	85.36	84.20	84.35	84.35	85.07
credit-g	70.50	74.00	69.60	69.90	67.70	73.10
dermatology	93.99	95.08	95.63	94.81	93.44	95.36
diabetes	73.83	74.09	72.40	70.05	69.92	74.35
echo	97.30	95.95	95.95	97.30	97.30	97.30
ecoli-c	84.23	84.82	81.25	83.04	81.85	84.52
glass	66.82	71.03	74.30	65.89	64.95	70.09
haberman	71.90	74.84	72.55	68.30	71.24	74.51
heart-cleveland	77.56	79.21	82.18	77.23	74.26	79.87
heart-h	80.95	78.91	78.57	79.25	78.23	79.59
heart-statlog	76.67	80.00	80.37	77.41	75.19	80.00
hepatitis	83.87	83.23	85.81	79.35	81.29	83.87
horse-colic	85.33	85.60	83.42	83.15	83.15	86.14
hypothyroid	99.58	99.58	99.58	99.58	99.63	99.60
ionosphere	91.45	93.16	93.16	90.60	90.88	93.45
iris	96.00	95.33	93.33	93.33	94.00	95.33
kr-vs-kp	99.44	99.44	99.50	99.44	99.44	99.44
labor	73.68	84.21	89.47	77.19	82.46	84.21
led	100.00	100.00	100.00	100.00	100.00	100.00
lenses	83.33	79.17	70.83	70.83	79.17	79.17
liver-disorders	68.70	72.75	71.59	66.09	63.48	71.88
lungcancer	50.00	56.25	53.13	53.13	46.88	56.25
lymph	77.03	79.05	81.08	81.76	77.70	80.41
monks	96.53	100.00	100.00	100.00	100.00	100.00
mushroom	100.00	100.00	100.00	100.00	100.00	100.00
page-blocks	96.88	97.26	97.02	96.78	96.80	97.24
postop	70.00	68.89	56.67	62.22	58.89	67.78
primary-tumor	39.82	42.18	40.12	37.17	36.87	41.30
promoters	81.13	79.25	85.85	78.30	79.25	81.13
segment	96.93	97.40	98.48	97.01	96.36	97.45
sick	98.81	98.73	99.18	99.05	99.02	98.97
solar-flare	97.83	97.83	96.59	97.83	97.83	97.83
sonar	71.15	74.52	77.88	74.52	73.08	74.52
soybean	91.51	93.27	92.83	92.39	90.19	93.12
spect	78.28	80.52	80.15	77.53	77.90	79.03
tic-tac-toe	85.07	92.80	96.35	93.11	86.22	93.53
vehicle	72.46	76.60	76.24	72.81	73.40	76.48
vote	94.79	95.23	95.66	95.01	94.79	95.44
wine	93.82	94.94	96.63	94.38	93.82	95.51
yeast	56.00	60.78	56.40	54.58	55.73	60.51
zoo	92.08	94.06	96.04	91.09	90.10	93.07
Average:	82.37	83.99	83.62	81.84	81.36	84.02

that simply select the majority class in the training set (Weka's ZeroR learner). The shaded area of each graph represents the cases were BMC significantly outperforms bagging or boosting. As shown in Figure 8.5, if even one of the decision tree base classifiers is replaced with a weaker classifier, BMC achieves significantly higher average accuracy. Boosting does not become competitive again until all the base learners are replaced with weak learners. As shown in Figure 8.6, if over half of base classifiers are weak learners, the advantage BMC has over bagging in terms of average accuracy also reaches statistical significance. Like boosting, bagging only becomes competitive again once all the base learners are weak learners. If the strength of the base classifiers is unbalanced, BMC often has the advantage over the *ad hoc* methods.

8.7 Bayesian Model Parameter Learning Given a Fixed Combination of Models

The previous experiments effectively use Bayesian techniques to determine the optimal combination of a fixed set of learners. Alternately, Bayesian techniques can be used to update learners given a fixed combination of weights. There are likely many models for which this sort of strategy could be applied, but one simple illustrative case involves the CMAC neural network topology [Albus, 1975].

The CMAC is modeled on the human cerebellum. It functions by mapping weights $w[i]$ to tiles which are interpreted spatially, as illustrated in Figure 8.7. Inputs are mapped to the correct bins by means of an association function $b[i](x)$, where $b[i](x) = 0$ when x does not fall within the spacial region assigned to bin i and where $b[i](x) = 1$ when it does. The output of the system can be computed as follows:

$$f_{CMAC}(x) = \sum_i w[i]b[i](x) \tag{8.6}$$

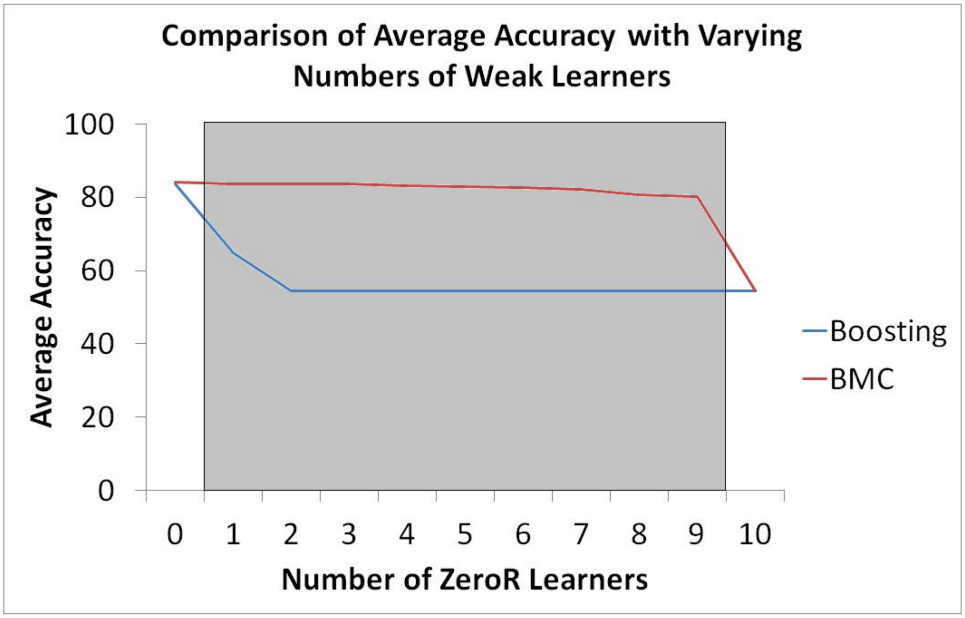


Figure 8.5: Average accuracy when a number of decision tree classifiers are replaced with weaker classifiers. Shaded area represents the cases where BMC significantly outperforms boosting.

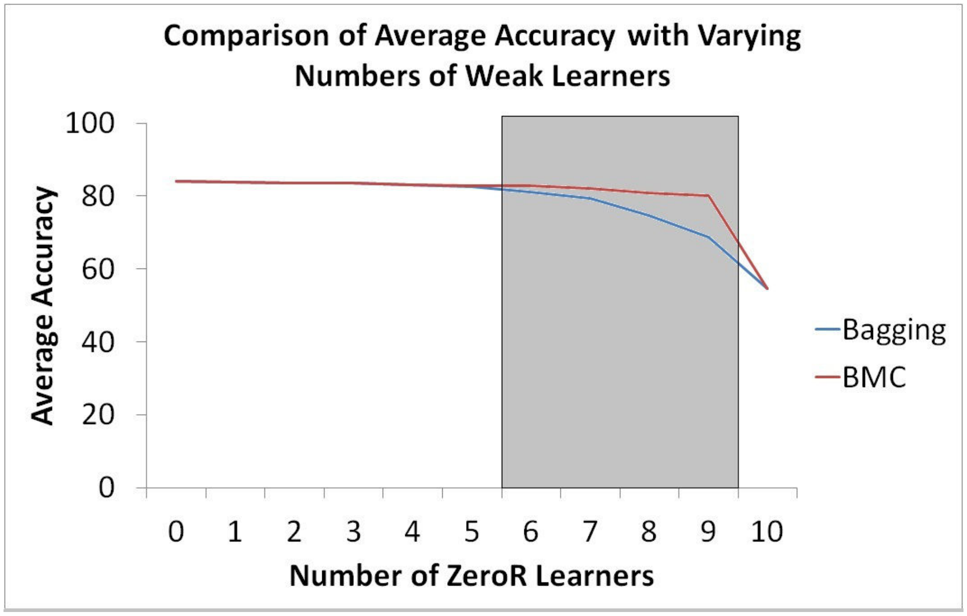


Figure 8.6: Average accuracy when a number of decision tree classifiers are replaced with weaker classifiers. Shaded area represents the cases where BMC significantly outperforms bagging.

Note that the CMAC outputs continuous values, so the experiments in this section will involve data sets with real rather than discrete target values. The error at location x is calculated as shown:

$$e(x) = f_{CMAC}(x) - f_{observed}(x) \tag{8.7}$$

Traditionally, weights are updated as follows:

$$\Delta w[i] = \alpha \frac{e(x)}{\sum_i b[i](x)} \tag{8.8}$$

where α is the learning rate. The output y of the network at any position x is the sum of the weights for the tiles that overlap that position.

Though not a traditional view, the CMAC can be thought of as an ensemble where each layer learns information about a given function and outputs are calculated by combining information from each layer using a fixed weighting scheme (each layer is equally weighted with all the others). The ensemble-like structure suggests that the CMAC could also be reasonably trained using ensemble techniques such as bagging or Bayesian model averaging, treating the layers as individual learners and altering the weightings of layer outputs according to the given technique. With one task specifically designed to match the assumptions made by BMA, that ensemble creation technique is effective in reducing error. However, once again, a Bayesian strategy that allows for a model combination approach does better on a wider variety of tasks.

The CMAC has an ensemble-like structure, and the posterior distribution over its parameters can be solved in closed form. Notice here that the CMAC weights for each layer are not the ensemble weights, but rather form the parameters of the individual component learners.

Carroll et al. [2007] showed how Bayesian techniques can be applied to CMAC learning. Further details on BCMAC training can be found elsewhere in the literature [Carroll,

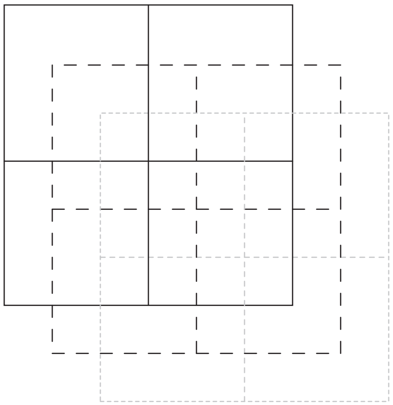


Figure 8.7: Tile structure for a CMAC with three layers and four tiles per layer

2010], but a brief overview is provided here. A function, f, is assumed to be stationary, and all observations y are assumed to have linear Gaussian noise with covariance Σ_y. The relationship between the data D and the CMAC's representation for f can be modeled as follows:

$$p(\mathbf{y}|\mathbf{x}, f) = N(\mathbf{y}; f(\mathbf{x}), \boldsymbol{\Sigma}_y). \tag{8.9}$$

This can be rewritten as:

$$p(\mathbf{y}|\mathbf{x}, f) = N(\mathbf{y}|\mathbf{H}\mathbf{w}, \boldsymbol{\Sigma}_y), \tag{8.10}$$

where $\mathbf{H}$ can be thought of as an association matrix. $\mathbf{H}_{i,j} = 1$ if tile j influences the training example i. Weight values are represented by the vector $\mathbf{w}$.

Weights of the model are related to observations according to a multivariate normal model [DeGroot, 1970] with prior parameters $\boldsymbol{\mu}_{\mathbf{0}}$ and $\boldsymbol{\Sigma}_0$. The parameters of the posterior distributions for the mean and covariance can then be found by:

$$\boldsymbol{\mu}_1 = \boldsymbol{\mu}_0 + \mathbf{K}_1(\mathbf{y} - \mathbf{H}\boldsymbol{\mu}_0), \tag{8.11}$$

and

$$\mathbf{\Sigma}_1 = (\mathbf{I} - \mathbf{K}_1\mathbf{H})(\mathbf{\Sigma}_0), \tag{8.12}$$

where

$$\mathbf{K}_1 = (\mathbf{\Sigma}_0)\mathbf{H}^T(\mathbf{H}(\mathbf{\Sigma}_0)\mathbf{H}^T + \mathbf{\Sigma}_y)^{-1}. \tag{8.13}$$

These equations are identical to the Kalman filter for a single time step. This observation means that, given a prior over CMAC weights and some training data, a well-known and widely studied filtering technique can be applied to solve in closed form for both the posterior distribution over the CMAC weights and the posterior predictive distribution over CMAC outputs.

The benefits of this strategy are demonstrated in the following experiments. The layers of the CMAC were learned using the traditional CMAC learning rule, bagging, Bayesian model averaging, and the BCMAC learning rule. All of the CMACs were constructed with five layers and between three and seven tiles per dimension on each layer. With the bagging CMAC, layers were trained individual on size n subsets selected with replacement from the initial training set of size n. Outputs of each layer were then weighted equally when calculating the final output for a given example. The Bayesian model averaging CMAC was constructed in a similar manner, but layer outputs were weighted by a likelihood term calculated using a normal noise model. Priors for the BCMAC were calculated empirically based on the data sets.

Experiments are conducted on three numeric data sets provided by Weka for machine learning tasks [Witten and Frank, 2005]. Because the CMAC was designed for continuous values, these sets were selected for their limited number of numerical features and numeric class values. Algorithm performance was also tested on twoDimEgg, a variant of the two-dimensional egg carton function $y = sin(x_1 * 2.5) + sin(x_2 * 2.5)$, and step2d, a stepwise function which returns 1 if $x_1^2 + x_2^2 < 10$ and -1 otherwise. This rather simple function was

Table 8.10: Average error rates of four learning strategies

	CMAC	Bagging	BMA	BCMAC
elusage	0.047	0.045	0.045	0.035
gascon	0.140	0.135	0.134	0.041
longley	0.097	0.119	0.119	0.062
step2d	0.019	0.018	0.022	0.018
twoDimEgg	0.025	0.109	0.270	0.018
optimalBMA	0.005	0.071	0.006	0.002

specifically chosen to have a steep, curved boundary, a situation which have been shown to be difficult for CMAC based learning algorithms.

In order to further test the theory that BMA performs poorly because it performs optimal model selection instead of optimal model combination, the final data set, optimalBMA, was constructed to provide a situation where model selection would perform well. The function assigns -1 to all values left of a vertical boundary and 1 to all values to the right. This boundary was aligned with the edge of one of the tiles in the CMAC. Thus, one of the layers would provide correct outputs for each example and every other layer would provide at least some incorrect outputs, and the goal of the ensemble would be to select this layer.

The BCMAC achieves a substantially lower error rate than the Bayesian model averaging strategy on all data sets studied, except for the case of optimalBMA where the results are nearly indistinguishable. In fact, with the exception of one tie with bagging on the step2d function, BCMAC outperforms all of the other three algorithms in terms of error reduction over the other five data sets. As with the previous experiments, bagging was often able to achieve a lower error rate than Bayesian model averaging. However, Bayesian model averaging substantially outperforms bagging on the optimalBMA data set, where placing all of the weight on one component is the best strategy. BMA was outperformed by the *ad hoc* techniques, except in the one case where model selection was required. This again provides further empirical justification for Minka's proposition on the theory of ensemble learning.

8.8 Conclusion

Despite the theoretical optimality of Bayesian methods and their successful application to a wide variety of tasks, the standard technique of Bayesian model averaging struggles in empirical studies. Minka theorized that since the algorithm places so much emphasis on the most likely ensemble member, it fails to take advantage of the benefits inherent in model combinations. However, if BMA is modified to integrate over combinations of models rather than over individual learners, it can achieve much better results.

Domingos described a number of situations in which Bayesian model averaging is outperformed by standard *ad hoc* ensemble creation methods. We have shown that even the most simplistic of Bayesian model combination strategies outperforms the traditional *ad hoc* techniques of bagging and boosting, as well as outperforming BMA in a significant number of cases. We have demonstrated with the BCMAC experiments that, in the rare instances where model selection is indeed the correct approach, Bayesian model averaging performs well. On most problems, however, a Bayesian technique geared toward selecting a combination of models results in lower error rates.

This work has some theoretical implications for why ensembles work. The results suggest the effectiveness of ensembles is due, at least in part, to the enriched hypothesis space and more general bias that can be provided by a combination of models. We have demonstrated that there are a wide variety of potential methods for applying Bayesian techniques to model combination. We have shown that it is possible to fix the component learners and then learn the optimal model combination in a Bayesian fashion (both versions of BMC). We have also shown that in some situations it is possible to fix the model combination strategy, and learn optimal models given the known combination (BCMAC).

Future work will involve the investigation of more sophisticated methods of Bayesian model combination. For example, the simple Bayesian model combination strategies presented in Section 8.6 could be modified to allow for non-linear combinations of models. Other possible strategies might take spatial considerations into account, developing learners to specialize in different areas of the feature space or training learners with the sampling techniques used in boosting.

Alternately, strategies could be developed that employ an expectation maximization strategy. An optimal combination could be determined given a set of learners, and then the learners could be updated given the new combination strategy. Of particular interest are strategies that would allow learners and combinations to be determined simultaneously using Bayesian techniques. The BCMAC can be solved in closed form because both weights and outputs are distributed normally. Other learners with similar Normal distribution properties might also be solved in a similar fashion.

Part IV

Conclusion

Computers and electronic music are not the opposite of the warm human music. It's exactly the same. **–Bill Laswell**

Chapter 9

Contributions and Future Work

***Remember always that the composer's pen is still mightier than the bow of the violinist; in you lie all the possibilities of the creation of beauty.* –John Philip Sousa**

This dissertation described a computational creative system capable of eliciting desired emotional and physiological responses, often at a level similar to that of human ability.

Chapter 2 discussed Colton's [2008] criteria of "skill," "appreciation," and "imagination" in evaluating the creativity. The system described in this work was able to demonstrate its "skill" and "appreciation" by generating music that tends to behave according to traditional musical conventions and accurately convey particular emotions. Not surprisingly, the human-generated songs were rated as more musical on average (7.81 compared to 6.73 on a scale of 1 to 10), but a number of the individual computer-generated selections were rated more musical than some of the individual human-composed selections. According to survey data, 54% of respondents correctly identified the target emotion in the computer-generated songs, while only 43% of respondents did so for human-generated songs. The system demonstrates its "imagination" by generating original compositions that were rated by listeners as being fairly unique. Computer-generated selections received an average rating of 4.86 for novelty compared to a 4.67 for human-composed selections.

Chapter 3 discussed how the system models various characteristics that contribute to the emotional content of music and reports the results of surveys taken on a larger number of computer-generated works. 58% of respondents correctly identified the intended emotion in computer-generated selections as compared to 33% for human-composed ones. When considering unconstrained responses, percentages of subjects identifying the intended emotions were 22% and 17% respectively for computer-generated and human-composed selections.

Chapter 4 extended the function of the system to generating selections that elicit particular physiological responses. When compared to human performance, experiments demonstrated that the system was equally adept at eliciting changes in skin temperature and heart rate and more effective at eliciting changes in breathing rate and skin resistance. The system is particularly adept at composing pieces that elicit target responses in individuals who demonstrated predictable responses to training selections.

Chapters 5 and 6 demonstrated practical applications of the system. In Chapter 5, when music with targeted emotional content was paired with the emotion-labeled text of fairy tales, it made the stories significantly more enjoyable to listen to and increased listener perception of emotion in the text. On a scale of 1 to 5, average ratings for listener enjoyment were 3.08 for text without music, 2.78 for text paired with music with random emotional content, and 3.43 for text paired with music with targeted emotional content. Average intensity ratings for perceived emotions in the stories were 1.85 for the "no music" option, 1.88 for "random music," and 2.10 for "emotionally targeted music."

When the system was used to generate melodic accompaniment for lyrics in Chapter 6, it was often able to do so at a level similar to that of human ability in terms of melodic pleasantness and lyric/note fit. For example, melodies generated in the "bluegrass" style received an average score of 3.52 for melodic pleasantness (again on a scale of 1 to 5). The average for the original tunes was 3.51. In addition, the system was able to generate some "surprising" but effective selections, ones where listeners gave the tunes low ratings for style expectedness but high ratings for melodic pleasantness.

Chapter 7 presented the strategy of "Aggregate Certainty Estimators," a technique that combines votes of an ensemble by using multiple measures to estimate a classifier's certainty in its prediction for a given instance. This technique is able to outperform the strategies of bagging, boosting, “SelectBest,” arbitration, and modified stacking in terms of average classifier accuracy over 36 data sets.

Chapter 8 discussed the pitfalls of another popular ensemble creation strategy, Bayesian model averaging. It proposed a novel technique, “Bayesian model combination” which was able to significantly outperform Bayesian model averaging in terms of average accuracy over 50 data sets. Under the right conditions, it was also able to significantly outperform bagging, boosting, and stacking in terms of average classification accuracy.

Possibilities for future work include refining the parameters of the music-generating system. For example, different values of n for the n-gram models would likely produce different results, which could be analyzed for comparative pleasantness and originality. Similar experiments could also be conducted by varying the number of songs in a training corpus. More musical selections might also be achieved through refinements to the rhythm and pitch evaluators. Further additions to the system might include more variations in musical form and extension of the length of generated works.

As previously mentioned, the system borrows heavily from the accompaniment patterns of the training corpus. Further work could involve the analysis of an array of standard MIDI accompaniment files to determine their effectiveness at eliciting particular emotional and physiological responses. These could then be used to provide more generic accompaniments for the generated selections.

Future work might also involve finding more applications for the system. For example, it could be used in conjunction with other creative systems to provide accompaniments for automatically generated stories or games. Compositions could also be tailored to individuals. Given the variation in subject response, particularly with physiological measures, it would be interesting to analyze the system’s effectiveness in generating music targeted to elicit a given response in a given individual by using training selections that are also person-specific.

References

J. S. Albus. A new approach to manipulator control: The cerebellar model articulation controller (CMAC). *Journal of Dynamic Systems, Measurement, and Control*, 97(3):220–227, 1975.

K. M. Ali and M. J. Pazzani. Error reduction through learning multiple descriptions. *Machine Learning*, 24(3):173–202, 1996.

S. O. Ali and Z. F. Peynircioglu. Songs and emotions: Are lyrics and melodies equal partners? *Psychology of Music*, 4(4):511–534, 2006.

M. Allan and C. K. I. Williams. Harmonising chorales by probabilistic inference. *Advances in Neural Information Processing Systems*, 17:25–32, 2005.

K. Allen and J. Blascovich. Effects of music on cardiovascular reactivity among surgeons. *Journal of the American Medical Association*, 272(11):882–884, 1994.

American Heart Association. All about heart rate, 2012. URL `http://www.heart.org/`.

K. Ang, S. Yu, and E. Ong. Theme-based cause-effect planning for multiple-scene story generation. In *Proceedings of the International Conference on Computational Creativity*, pages 48–53, 2011.

J. Bates. The role of emotion in believable agents. *Communications of the ACM*, 37(7): 122–125, 1994.

L. Bernardi, C. Porta, and P. Sleight. Cardiovascular, cerebrovascular, and respiratory changes induced by different types of music in musicians and non-musicians: The importance of silence. *Heart*, 92:445–452, 2006.

J. M. Bernardo and A. F. M. Smith. *Bayesian Theory.* Wiley, New York, NY, 1994.

D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

A. J. Blood and R. J. Zatorre. Intensely pleasurable responses to music correlate with activity in brain regions implicated in reward and emotion. In *Proceedings of the National Academy of Sciences*, volume 98(20), pages 11818–11823, 2001.

M. Boden. Creativity and unpredictability. *Stanford Humanities Review*, 4(2):123–139, 1995.

M. G. Boltz. The cognitive processing of film and musical soundtracks. *Memory and Cognition*, 32(7):1194–1205, 2004.

M. M. Bradley and P. J. Lang. Affective norms for english words ANEW: Stimuli, instruction manual and affective ratings. Technical Report c-1, The Center for Research in Psychophysiology, University of Florida, 1999.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

M. D. Brownell. Musically adapted social stories to modify behaviors in students with autism: Four case studies. *Journal of Music Therapy*, 39:117–144, 2002.

C. Bullerjahn and M. Guldenring. An empirical investigation of effects of film music using qualitative content analysis. *Psychomusicology*, 13:99–118, 1994.

J. G. Carney, P. Cunningham, and U. Bhagwan. Confidence and prediction intervals for neural network ensembles. In *International Joint Conference on Neural Networks*, volume 2, pages 1215–1218. IEEE, 1999.

J. L. Carroll. *A Bayesian Decision Theoretical Approach to Supervised Learning, Selective Sampling, and Empirical Function Optimization.* PhD thesis, Brigham Young University, March 2010. URL `http://james.jlcarroll.net/publications/`.

J. L. Carroll, C. K. Monson, and K. D. Seppi. A Bayesian CMAC for high assurance supervised learning. *Applications of Neural Networks in High-Assurance Systems, IJCNN Workshop*, 2007.

W. Chai and B. Vercoe. Folk music classification using hidden Markov models. In *Proceedings of the International Conference on Artificial Intelligence*, 2001.

C. Chuan and E. Chew. A hybrid system for automatic generation of style-specific accompaniment. In *Proceedings of the International Joint Workshop on Computational Creativity*, pages 57–64, 2007.

A. J. Cohen. Music as a source of emotion in film. *Music and emotion: Theory and research*, pages 249–272, 2001.

S. Colton. Creativity versus the perception of creativity in computational systems. In *Creative Intelligent Systems: Papers from the AAAI Spring Symposium*, pages 14–20, Stanford, CA, 2008. AAAI Press.

S. Colton, A. Pease, and J. Charnley. Computational creativity theory: The FACE and IDEA descriptive models. In *Proceedings of the 2nd International Conference on Computational Creativity*, pages 90–95, 2011.

D. Conklin. Music generation from statistical models. In *Proceedings of the AISB Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, pages 30–35, 2003.

D. Cope. *Computer Models of Musical Creativity.* The MIT Press, Cambridge, Massachusetts, 2006.

T. M. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27, 1967.

L. J. Cronbach. Coefficient alpha and the internal structure of tests. *Psychometrika*, 16(3): 297–334, 1951.

M. Csikszentmihalyi. *Creativity: Flow and the Psychology of Discovery and Invention.* Harper Perennial, New York, 1996.

R. B. Dannenberg. An on-line algorithm for real-time accompaniment. In *Proceedings of the International Computer Music Conference*, pages 279–289, 1985.

R. B. Dannenberg, W. P. Birmingham, G. Tzanetakis, C. Meek, N. Hu, and B. Pardo. The MUSART testbed for query-by-humming evaluation. In *Proceedings of the International Conference on Music Information Retrieval*, pages 41–51, 2003.

A. O. de la Puente, R. S. Alfonso, and M. A. Moreno. Automatic composition of music by means of grammatical evolution. In *Proceedings of the International Conference on APL*, pages 148–155, New York, 2002. ACM Press.

M. H. DeGroot. *Optimal Statistical Decisions.* McGraw-Hill Book Company, New York, NY, 1970.

M. Delgado, W. Fajardo, and M. Molina-Solana. Inmamusys: Intelligent multi-agent music system. *Expert Systems with Applications*, 36(3-1):4574–4580, 2009.

K. B. Dickerson and D. Ventura. Music recommendation and query-by-content using self-organizing maps. In *Proceedings of the International Joint Conference on Neural Networks*, pages 705–710, 2009.

S. Dolev, G. Leshem, and R. Yagel. Purifying data by machine learning with certainty levels. In *Proceedings of the Third International Workshop on Reliability, Availability, and Security*. ACM, 2010.

P. Domingos. Bayesian averaging of classifiers and the overfitting problem. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 223–230, 2000.

P. Domingos and M. J. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.

S. Dzeroski and B. Zenko. Is combining classifiers with stacking better than selecting the best one? *Machine Learning*, 54:255–273, 2004.

S. Dzeroski, B. Cestnik, and I. Petrovski. Using the m-estimate in rule induction. *Journal of Computing and Information Technology*, 1:37–46, 1993.

C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.

C. Ferri, P. Flach, and J. Hernandez-Orallo. Improving the AUC of probabilistic estimation trees. In *Proceedings of the Fourteenth European Conference of Machine Learning*, pages 121–132, 2003.

C. Ferri, P. Flach, and J. Hernandez-Orallo. Delegating classifiers. In *Proceedings of the Twenty-first International Conference on Machine Learning*, pages 289–296, 2004.

V. Francisco and R. Hervás. EmoTag: Automated mark up of affective information in texts. In *EUROLAN Summer School Doctoral Consortium*, pages 5–12, Iasi, Romania, July 2007. ISBN 978-973-703-246-1.

Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148–156, 1996.

A. Gabrielsson and E. Lindstrom. The influence of musical structure on emotional expression. *Music and Emotion: Theory and Research*, pages 223–248, 2001.

P. Gervás. An expert system for the composition of formal spanish poetry. *Journal of Knowledge-Based Systems*, 114(3-4):181–188, 2001.

P. Gervás, B. Díaz-Agudo, F. Peinado, and R. Hervás. Story plot generation based on CBR. *Journal of Knowledge-Based Systems*, 18(4–5):235–242, 2005.

K. Gfeller. Music, the language of emotions. In R.F. Unkefer, editor, *Music Therapy in the Treatment of Adults with Mental Disorders; Theoretical Basis and Clinical Interventions*, pages 42–59. Schirmer Books, New York, 1990.

W. R. Gilks. Markov chain monte carlo. *Encyclopedia of Biostatistics*, 2005.

F. He and D. Xiaoqing. Improving naive Bayes text classifier using smoothing methods. *Advances in Information Retrieval*, pages 703–707, 2007.

S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases, 1998. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

T. K. Ho, J. J. Hull, and S. N. Srihari. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):66–75, 1994.

J. Hoeting, D. Madigan, A. Raftery, and C. Volinsky. Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–417, 1999.

A. Horner and D. Goldberg. Genetic algorithms and computer-assisted music composition. In *Proceedings of the International Conference on Genetic Algorithms*, pages 479–482, Urbana-Champaign, Illinois, 1991.

I-330-C2+ Hardware Guide, 2004.

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.

G. H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345, San Francisco, CA, 1995. Morgan Kaufmann.

W. Johnson. Face the music. *Film Quarterly*, 22:3–19, l969.

P. N. Juslin. Communicating emotion in music performance: A review and a theoretical framework. *Music and Emotion: Theory and Research*, pages 223–248, 2001.

S. Khalfa, I. Peretz, J. Blondin, and R. Manon. Event-related skin conductance responses to musical emotions in humans. *Neuroscience Letters*, pages 145–149, 2002.

P. Kivy. *The Corded Shell: Reflections on Musical Expression.* Princeton University Press, Princeton, NJ, 1980.

J. Klein, Y. Moon, and R. Picard. This computer responds to user frustration: Theory, design, results, and implications. *Interacting with Computers*, 14:119–140, 2002.

R. Kohavi. The power of decision tables. In *Proceedings of the Eighth European Conference of Machine Learning*, pages 174–189, 1995.

K. Koskenniemi. A general computational model of word-form recognition and production. In *Proceedings of the Tenth International Conference on Computational Linguistics*, pages 178–181, Stroudsburg, PA, 1984. Association for Computational Linguistics.

H. Kreitler and S. Kreitler. *Psychology of the Arts.* Duke University Press, Durham, NC, 1972.

K. Lang. NewsWeeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, 1995.

C. Lepage, P. Drolet, M. Girard, Y. Grenier, and R. DeGagne. Music decreases sedative requirements during spinal anesthesia. *Anesthesia-Analgesia*, 93:912–916, 2001.

G. Lewis. Too many notes: Computers, complexity and culture in voyager. *Leonardo Music Journal*, 10:33–39, 2000.

T. Li and M. Ogihara. Content-based music similarity search and emotion detection. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, V705-V708:17–21, 2004.

L. Macedo. Creativity and surprise. In *Proceedings of the AISB Symposium on Artificial Intelligence and Creativity in Arts and Science*, pages 84–92, York, UK: University of York, 2001.

L. Macedo and A. Cardoso. Assessing creativity: The importance of unexpected novelty. In *Proceedings of the ECAI Workshop on Creative Systems: Approaches to Creativity in Artificial Intelligence and Cognitive Science*, pages 31–38, 2002.

S. Marshall and A. J. Cohen. Effects of musical soundtracks on attitudes toward animated geometric figures. *Music Perception*, 6:95–112, l988.

P. McCorduck. *Machines Who Think.* A. K. Peters, Ltd., Natick, MA, 2nd edition, 2004. ISBN 1-56881-205-1.

C. McKay and I. Fujinaga. Automatic genre classification using large high-level musical feature sets. In *Proceedings of the Fifth International Symposium on Music Information Retrieval*, pages 525–530, 2004.

L. B. Meyer. *Emotion and Meaning in Music.* Chicago University Press, Chicago, 1956.

T. Minka. Bayesian model averaging is not model combination. MIT Media Lab Note, December 2000.

T. M. Mitchell. *Machine Learning.* WCB/McGraw-Hill, 1997.

K. Monteith, T. Martinez, and D. Ventura. Automatic generation of music for inducing emotive response. In *Proceedings of the First International Conference on Computational Creativity*, pages 140–149, 2010.

D. Norton, D. Heath, and D. Ventura. Establishing appreciation in a creative system. In *Proceedings of the First International Conference Computational Creativity*, pages 26–35, 2010.

M. Ochs, C. Pelachaud, and D. Sadek. An empathic virtual dialog agent to improve human-machine interaction. In *Proceedings of the Seventh International Joint Conference on Autonomous Agent and Multi-Agent Systems*, pages 89–96, 2008.

A. Ohman. Preattentive processes in the generation of emotions. *Cognitive Perspectives on Emotion and Motivation*, pages 127–144, 1988.

A. Oliveira and A. Cardoso. Towards affective-psychophysiological foundations for music production. In *Proceedings of the 2nd International Conference on Affective Computing and Intelligent Interaction*, pages 511–522, 2007.

J. Ortega, M. Koppel, and S. Argamon. Arbitrating among competing classifiers using learned referees. *Knowledge and Information Systems Journal*, 3(4):470–490, 2001.

E. Pampalk, A. Flexer, and G. Widmer. Improvements of audio-based music similarity and genre classification. In *Proceedings of the Sixth International Symposium on Music Information Retrieval*, pages 623–633, 2005.

W. G. Parrott. *Emotions in Social Psychology.* Psychology Press, Philadelphia, 2001.

T. Partala and V. Surakka. The effects of affective interventions in human-computer interaction. In *Proceedings of the Conference on Human Factors in Computing Systems*, volume 16, pages 295–309, 2004.

M. T. Pearce and G. A. Wiggins. Evaluating cognitive models of musical composition. In *Proceedings of the 4th International Joint Workshop on Computational Creativity*, pages 73–80, 2007.

M. T. Pearce, D. Meredith, and G. A. Wiggins. Motivations and methodologies for automation of the compositional process. *Musicae Scientiae*, 6(2):119–147, 2002.

E. Peper, H. Tylova, K.H. Gibney, R. Harvey, and D. Combatalade. *Biofeedback mastery-An experiential teaching and self-training manual.* AAPB, Wheat Ridge, CO, 2008.

R. Pérez y Pérez and M. Sharples. Three computer-based models of storytelling: BRUTUS, MINSTREL and MEXICA. *Knowledge-Based System*, 17(1):15–29, 2004.

S. Phon-Amnuaisuk and G. Wiggins. The four-part harmonization problem: A comparison between genetic algorithms and a rule-based system. In *Proceedings of the AISB Symposium on Musical Creativity*, pages 28–34, Edinburgh, 1999.

R. W. Picard. Affective computing. Technical Report 321, MIT, 1995.

D. Ponsford, G. Wiggins, and C. Mellish. Statistical learning of harmonic movement. *Journal of New Music Research*, 28(2):150–177, 1998.

F. Provost and P. Domingos. Tree induction for probability-based ranking. *Machine Learning*, 52:199–216, 2003.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufman, 1993.

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77(2), pages 257–286, 1989.

F. Rahman and R. Manurung. Multiobjective optimization for meaningful metrical poetry. In *Proceedings of the Second International Conference on Computational Creativity*, pages 4–9, 2011.

M. Richard and R. Lippman. Neural network classifiers estimate Bayesian a-posteriori probabilities. *Neural Computation*, 3:461–483, 1991.

N. S. Rickard. Intense emotional responses to music: A test of the phyisological arousal hypothesis. *Psychology of Music*, 32(4):371–388, 2004.

M. Riedl. *Narrative Generation: Balancing Plot and Character*. PhD thesis, North Carolina State University, 2004.

L. Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1):1–39, 2010.

D. W. Ruck, S. K. Rogers, M. Kabrisky, M. E. Oxley, and B. W. Suter. The multilayer perceptron as an approximation to a Bayes optimal discriminant function. *IEEE Transactions on Neural Networks*, 1(4):296–298, 1990.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, 1:318–362, 1986.

J. Rutherford and G. Wiggins. An experiment in the automatic creation of music which has specific emotional content. In *Proceedings of MOSART, Workshop on Current Research Directions in Computer Music*, pages 35–40, Barcelona, Spain, 2003.

A. P. Saygin, I. Cicekli, and V. Akman. Turing test: 50 years later. *Minds and Machines*, 10(4):463–518, 2000.

S. Schachter and J. Singer. Cognitive, social, and physiological determinants of emotional state. *Psychological Review*, 69:379–399, 1962.

R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 80–91, 1998.

R.E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

K. R. Scherer. On the nature and function of emotion: A component process approach. In *Approaches to Emotion*, pages 293–317. Lawrence Erlbaum Associates, Hillsdale, NJ, 1984.

K. R. Scherer. On the symbolic functions of vocal affect expression. *Journal of Language and Social Psychology*, 7:79–100, 1988.

K. R. Scherer. Emotional effects of music: Production rules. In *Music and Emotion: Theory and Research*, pages 223–248. Oxford University Press, Oxford, U.K., 2001.

G. Schlaug, S. Marchina, and A. Norton. From singing to speaking: Why patients with Broca's aphasia can sing and how that may lead to recovery of expressive language functions. *Music Perception*, 25:315–323, 2008.

J. A. Sloboda. *The Musical Mind: The Cognitive Psychology of Music.* Oxford University Press, Oxford, 1985.

J. Thayer and R. Levenson. Effects of music on psychophysiological responses to a stressful film. *Psychomusicology*, 3:44–54, l983.

N. Tokui and H. Iba. Music composition with interactive evolutionary computation. In *Proceedings of the Third International Conference on Generative Art*, pages 215–226, Milan, Italy, 2000.

E. Tolbert. Music and meaning: An evolutionary story. *Psychology of Music*, 24:103–130, 2001.

G. J. Tortora and N. P. Anagnostakos. *Principles of Anatomy and Physiology, 6th edition.* Harper-Collins, New York, 1990.

Y. Wang, M.-Y. Kan, T. L. Nwe, A. Shenoy, and J. Yin. LyricAlly: Automatic synchronization of acoustic musical signals and textual lyrics. In *Proceedings of the 12th Annual ACM International Conference on Multimedia*, pages 212–219, 2004.

J. Weizenbaum. ELIZA - a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

J. M. White. Effects of relaxing music on cardiac autonomic balance and anxiety after acute myocardial infarction. *American Journal of Critical Care*, 8:220–230, 1999.

G. Wiggins. A preliminary framework for description, analysis and comparison of creative systems. *Journal of Knowledge Based Systems*, 19(7):449–458, 2006.

G. A. Wiggins, M. T. Pearce, and D. Mullensiefen. Computational modelling of music cognition and musical creativity. In R. Dean, editor, *Oxford Handbook of Computer Music and Digital Sound Culture*, pages 383–420. Oxford University Press, 2009.

T. Wigram. Indications in music therapy: Evidence from assessment that can identify the expectations of music therapy as a treatment for autistic spectrum disorder (ASD): Meeting the challenge of evidence based practice. *British Journal of Music Therapy*, 16: 11–28, 2002.

I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, San Francisco, 2nd edition, 2005.

D Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–260, 1992.

Y.-S. Wu, W. r. Chu, C.-Y. Chi, D. C. Wu, R. T.-H. Tsai, and J. Y. j Hsu. The power of words: Enhancing music mood estimation with textual input of lyrics. In *Proceedings of the International Conference on Affective Computing and Intelligent Interaction*, pages 1–6, 2009.

M. R. Zentner and J. Kagan. Perception of music by infants. *Nature*, 383(29):1–16, 1996.