BAYESIAN APPROACHES TO MUSICAL INSTRUMENT CLASSIFICATION USING TIMBRE SEGMENTATION

by

Patrick Joseph Donnelly

A dissertation proposal submitted in partial fulfillment
of the requirements for the degree

of

Doctor of Philosophy

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

May, 2012

TABLE OF CONTENTS

TABLE OF CONTENTS – CONTINUED

ABSTRACT

The task of identifying musical instruments in an audio recording is a difficult problem. While there exists a body of literature on single instrument identification, little research has been performed on the more complex, but real-world, situation of more than one instrument present in the signal. This work proposes a Bayesian method for multi-label classification of musical instrument timbre.

Preliminary results demonstrate the efficacy of Bayesian networks on the single instrument classification problem. Peak spectral amplitude in ten frequency windows were extracted for each of twenty time windows to be used as features. Over a dataset of 24,000 audio examples covering the full musical range of 24 different common orchestral instruments, four different Bayesian network structures, including naïve Bayes, were examined and compared to two support vector machines and a k-nearest neighbor classifier. Classification accuracy was examined by instrument, instrument family, and dataset size. Bayesian networks with conditional dependencies in the time and frequency dimensions achieved 98% accuracy in the instrument classification task and 97% accuracy in the instrument family identification task. These results demonstrated a significant improvement over the previous approaches in the literature on this dataset.

The remainder of this proposal outlines my approach for the identification of musical instrument timbre when more than one instrument is present in the signal. First, signature matching Bayesian networks will be trained on single instruments to recognize the timbral signature of individual instruments. Secondly, those signatures will be used to extract the features relevant to a single instrument from the spectral analysis of a multi-instrument signal. Finally, a binary-relevance Bayesian classifier will determine if each specific instrument is present in the signal.

This system proposes a novel approach to template matching allowing for probabilistic segmentation of musical spectra. Furthermore the proposed approaches outline a novel approach to multi-label classification of music instrument timbre which supports both harmonic and inharmonic instruments, scales to a large number of musical instruments, and allows for efficient classification of new examples given the trained models.

CHAPTER 1
INTRODUCTION

The ability of a computer to learn to an identify the musical instruments present in audio recording is an important problem within the field of Music Information Retrieval (MIR). For instance, Digital Media Stores, such as iTunes or Amazon, or Recommendation Systems, such as Pandora, might wish to automatically categorize their music catalog, allowing search and retrieval by specific musical instrument. Timbre identification is also an important task in the area of musical genre categorization, automatic score creation, and audio track separation.

The identification of musical instruments in audio recordings is a frequently explored, yet unsolved, classification problem. Most approaches in the literature have focused on the identification of a single musical instrument. Despite a number of experiments in the literature over the years, no single feature extraction scheme or learning approach has emerged as a definitive solution to this classification problem. The most common approaches in the literature are the k-nearest neighbor algorithm and the support vector machine (SVM).

In recent years, investigation has turned towards the identification of multiple instruments present in a recording. The goal of musical instrument classification is to determine the instruments present in audio recordings and automatically assign the appropriate metadata labels. This process would occur offline. A MIR system could then, in real-time, retrieve all audio recordings that contain a specific queried musical instrument from a large dataset of recordings.

This work proposes a Bayesian approach to the classification problem of musical instrument timbre, using timbre segmentation and focusing on the task of multi-label classification. Chapter 2 explains musical timbre and reviews the relevant algorithms

described in this proposal. Chapter 3 reviews the relevant literature of both multi-class and multilabel classification of musical timbre. Chapter 4 introduces a feature extraction scheme based on a psychoacoustic definition of timbre and demonstrates the validity of a Bayesian approach to the single instrument classification problem. Lastly, Chapter 5 outlines my proposal for a Bayesian approach to the identification of multiple musical instrument present in the same signal.

CHAPTER 2
BACKGROUND WORK

2.1 Timbre

When a musical instrument plays a note, we perceive both a musical pitch and the instrument playing that note. Timbre, or tone color, is the psychoacoustic property of sound that allows the human brain to distinguish readily between the same note, even when played on two different instruments.

Overtones are the musical tones that are part of the harmonic series above a fundamental note we perceive. The primary musical pitch we perceive is the fundamental frequency. The fundamental frequency and its overtones are collectively known as partials. Harmonic instruments are those whose partials are approximate integer multiples of the fundamental frequency. With the exception of drums and bells, such as chimes, most orchestral instruments are harmonic. The perception of timbre depends on the harmonics (spectra) and the fine timing (envelope) of each harmonic constituent (partial) of the musical signal [1].

2.2 Algorithms

The preliminary results in this work compare three types of algorithms on the machine learning task of timbre classification. This section briefly explains each of the algorithms described in this proposal.

2.2.1 Nearest Neighbor

The k-nearest neighbor (k-NN) algorithm is a common instance-based learning algorithm in which a previously unknown example is classified with the most common

class amongst its k nearest neighbors, where k is a small positive integer. A neighbor is determined by the application of some distance metric $D(\cdot,\cdot)$, such as Euclidean distance, in d multidimensional feature space. Formally, let $\mathcal{X}$ be a space of points where each feature vector $\mathbf{f} \in \mathcal{X}$ is defined as $\mathbf{f} = \langle \{f^1, \ldots, f^d\}; c \rangle$, c is the true class label, and $\mathcal{X}_{tr} \subset \mathcal{X}$ is the set of training examples. For a query example $\mathbf{f_q} \in \mathcal{X} - \mathcal{X}_{tr}$, 1-NN finds an example $\mathbf{f_r} \in \mathcal{X}_{tr}$ such that $\forall \mathbf{f_x} \in \mathcal{X}_{tr}$, $\mathbf{f_x} \neq \mathbf{f_r}$, $D(\mathbf{f_q}, \mathbf{f_r}) < D(\mathbf{f_q}, \mathbf{f_x})$ and returns the associated class label c_r [2]. When $k > 1$, the majority class among the set of k closest neighbors will be returned.

2.2.2 Support Vector Machine

The support vector machine (SVM) is a discriminant-based method for classification or regression. The SVM algorithm constructs a hyperplane in high dimensional space that represents the largest margin separating two classes of data. To support multiclass problems, the SVM is often implemented as a series of 'one-versus-all' binary classifiers. The SVM is defined as:

$$\min \frac{1}{2}\|w\|^2 + C \cdot \sum_i \xi_i \tag{2.1}$$

subject to:

$$y(\mathbf{w}^T \cdot \Phi(\mathbf{f}) + b) \leq 1 - \xi_i, \xi_i \geq 0 \tag{2.2}$$

where $\mathbf{f}$ is a vector of features, $\mathbf{w}^T$ is the discriminant vector, C is a regularizing coefficient, ξ_i is a slack variable, b is the bias offset, label $y \in \{-1, +1\}$, and the kernel function $K(\mathbf{f_i}, \mathbf{f_j}) = \Phi(\mathbf{f_i})^T \cdot \Phi(\mathbf{f_j})$ is the dot product of the basis function.

When the kernel function $K(\mathbf{f}) = \mathbf{f}$, the SVM is a linear classifier. When the kernel is a non-linear function, such as a polynomial (Equation 2.3), the features are

projected into a higher order space, which allows the algorithm to fit the maximum margin hyperplane in the transformed feature space, which is no longer linear in the original space [3].

$$K(\mathbf{f_i}, \mathbf{f_j}) = (\mathbf{f_i} \cdot \mathbf{f_j})^{\delta} \tag{2.3}$$

2.2.3 Bayesian Networks

Bayesian networks are probabilistic graphical models that are comprised of random variables, represented as nodes, and their conditional dependencies, represented as directed edges. The joint probability of the variables represented in the directed, acyclic graph can be calculated as the product of the individual probabilities of each variable, conditioned on each the node's parent variables. The Bayesian classifier without latent variables – hidden variables that are inferred rather than observed – is defined as:

$$\text{classify}(\mathbf{f}) = \underset{c \in C}{\text{argmax}}\, P(c) \prod_{f \in \mathbf{f}} P(f|\text{parent}(f)) \tag{2.4}$$

where $P(c)$ is the prior probability of class c and $P(f|\text{parent}(f))$ is the conditional probability of feature f given the values of that feature's parents. The classifier finds the class that has the highest probability of explaining the values of the feature vector [4].

CHAPTER 3

RELATED WORK

3.1 Single Instrument Classification

Beginning with initial investigations of music perceptionist John Grey [5], the task of musical instrument identification has relied on clustering techniques. Fujinaga created a k-NN system that achieved 68% instrument classification on a large database of 23 different recorded instruments [6].

In the 2000's, investigators began to explore other techniques. A seminal study using an SVM classified 200 milliseconds of recorded audio for eight musical instruments, using 16 Mel-frequency cepstral coefficients (MFCC) as features [7]. MFCC are coefficients of the power spectrum of a sound transformed along the mel scale of frequency. The authors achieved 70% accuracy using a 'one versus all' multi-class SVM with a polynomial kernel, which outperformed the 63% accuracy using Gaussian mixture models (GMM).

In 2003, a study demonstrated the ability of SVMs to outperform k-NN on the task of musical instrument identification. Agostini *et al.* used a set of nine spectral features and compared the results of an SVM, k-NN, and quadratic discriminant analysis (QDA). Their results on three different sets of instruments are shown in Table 3.1. For the 27 instrument set, the authors also tested instrument family discrimination (e.g., strings, woodwinds) and achieved 80.8% accuracy using an SVM compared to 76.2% using k-NN [8]. A more recent study used k-NN to achieve 93% instrument classification and 97% instrument family recognition on a set of 19 instruments [9].

Table 3.1: Results of [8]

Instruments	SVM	k-NN	QDA
17	**80.2**	73.5	77.2
20	**78.5**	74.5	75.0
27	**69.7**	65.7	68.5
Family	77.6	76.2	**80.8**

While k-NN and SVM remain the most commonly employed system for timbre classification, a few other approaches have been attempted. Kostek used a multilayer feedforward neural network to identify 12 musical instruments playing a wide variety of articulations using a combination of MPEG-7 and wavelet-based features. She achieved 71% accuracy, ranging from 55% correct identification of the English horn to 99% correct identification of the piano [10]. Like many other studies, Kostek noted the most common misclassification occurred between instruments within the same family and that performance deteriorated as the number of musical instruments increased. Another study employed a binary decision tree, a variation of the C4.5 algorithm, to classify 18 instruments using 62 features yielding 68% classification accuracy [11].

Despite a few attempts using other learning strategies, the focus in the literature remains dominated by SVM and k-NN for this task. While Bayesian networks, most commonly the hidden Markov model (HMM), have been widely used in the field of natural language processing for speech recognition, phoneme identification, and other tasks [12], Bayesian networks have not been widely used for the problem of musical instrument identification. One study presented preliminary results using a continuous-density HMM to classify seven different instrument groupings, but not individual instruments, achieving accuracies between 45% and 64% [13].

3.2 Multi-label Classification

In classification tasks, an instance $x \in \mathcal{X}$ is represented as an M-vector $x = [x_1, \ldots, x_M]$, where $\mathcal{X}$ denotes the attribute space. Single-label classification describes the assignment of instance x to a label l from a set of disjoint labels $\mathcal{L}$. In binary classification problems, $|\mathcal{L}| = 2$. In multi-class classification $|\mathcal{L}| > 2$, although each example is only assigned a single label.

In multi-label classification, on the other hand, each each example $x \in \mathcal{X}$ is assigned a set of labels $\mathcal{Y}$, where $\mathcal{Y} \subseteq \mathcal{L}$. Multi-label classification is increasingly popular in many areas, such as genre of films [14], text categorization [15], medical diagnosis [16], and classifying emotions in music [17].

There are three common approaches to multi-label classification. The first approach, known as algorithm extension, consists of adapting existing algorithms to return a set of labels instead of a single label. The second approach, transformation methods, describes the transformation of a multi-label classification problem into a single label multi-class problem. This is most often achieved by enumerating all possible sets of labels as if they were individual labels, which results in a combinatorial explosion in the number of labels [18]. For this reason, this approach is highly undesirable if $|\mathcal{L}|$ is a large number.

The third method is known as the Binary Relevance (BR) approach. The BR method learns $|\mathcal{L}|$ different binary classifiers, one for each possible label. Each binary classifier is trained to distinguish the examples in a single class from the examples in all remaining class. When classifying a new example, all $|\mathcal{L}|$ classifiers are run and the labels associated with the classifiers which output the label *true* are added to $\mathcal{Y}$. This is known as the one-vs-all (OVA) scheme. More specifically, each binary classifier C_l is responsible for predicting the true/false association for each single label $l \in \mathcal{L}$. The

final label set $\mathcal{Y}$ is the union of all labels from all classifiers that returned true [19]. This work will use the BR approach to multi-label classification.

3.3 Multi-label Instrument Classification

Only recently have investigators turned their attention to the problem of multi-label classification of instruments in polyphonic musical instrument signals. These studies often suffer from several limitations [20]:

Low accuracy: Accuracy is below 60% even for experiments using small datasets [18, 20, 21, 22].

Limited dataset: Small or limited datasets, often consisting of five or less instruments are used [22, 23, 24, 25] or datasets derived from synthetic MIDI instruments [26].

Predefined mixtures: Instrument sets are defined *a priori* and therefore the experiments cannot generalize to other sets of instruments [21].

Harmonic instruments: Approaches will work only for harmonic instruments and cannot be extended to inharmonic instruments such as drums and bells [20, 27].

Using a Gaussian Mixture Model (GMM), Eggink and Brown achieved 49% on pairs of instruments from a limited set of five instruments [24]. More recently, another study used GMM on a small dataset of five instruments to achieve 77% for two instrument mixtures, 43% for three instrument mixutres, and 40% for four instruments mixtures [22]. Using the transformation method, Essid and Richard achieved 53% accuracy using a GMM on a set of 20 instruments combinations ranging from single instruments up to combinations of four instruments [21].

Another study used k-NN to achieve 80% identification accuracy on the very limited set of eight instrumental mixtures [23]. More recently, another study compared an SVM and the Perceptron neural network to achieve 55% and 64% respectively on the identification of pairs of instruments from a large set of 25 harmonic instruments [20].

One study applied Linear Discriminant Analysis (LDA) on polyphonic training data to achieve 84% identification accuracy for two instrument mixtures, 77% for three instruments, and 72% for four instruments. However, the authors considered only the small dataset of five instruments and this approach would not scale well given a larger set of instruments in which it would not be feasible to generate training data for all permutations of many instruments [27].

CHAPTER 4
PRELIMINARY RESULTS

4.1 Overview

To demonstrate the validity of a Bayesian approach to musical instrument classification, preliminary experiments have been performed on the single instrument timbre classification problem. Using a feature extraction scheme based on a psychoacoustic definition of timbre, several different Bayesian network structures were compared to the baseline algorithms of support vector machines (SVM) and a k-nearest neighbor (k-NN) classifier. These results have been submitted for publication [28].

4.2 Data Generation

Our system uses 1000 audio examples for each musical instrument, covering 24 different orchestral instruments (Table 4.1). Each audio file is two seconds in duration, consisting of the instrument sustaining a single note for one second, and time before and after to capture the attack and the resonant decay, respectively. The audio samples were created using the EastWest Symphonic Orchestra sample library at the **MON**tana **ST**udio for **E**lectronics and **R**hythm (MONSTER) at Montana State University.

For each musical instrument, a MIDI control sequence was sent to a Kontakt Virtual Studio Technology (VST) player for rendering to audio. The resulting audio stream was recorded using the javax.sound package at a 44.1k sampling rate, 16-bits per sample, and stored as a single channel waveform audio file (WAV).

The pitch was randomly sampled uniformly with replacement covering the entire musical range of the instrument. The dynamic level was also sampled uniformly with

Table 4.1: Set of 24 instruments sorted by instrument family

Strings	Woodwinds	Brass	Percussion
Violin Viola Cello Contrabass Harp	Piccolo Flute Alto Flute Clarinet Bass Clarinet Oboe English Horn Bassoon Contrabassoon Organ	French Horn Trumpet Trombone Tuba	Chimes Glockenspiel Vibraphone Xylophone Timpani
5	10	4	5

replacement of the MIDI velocity parameter, covering the dynamic range pianissimo to fortissimo. In total, there are 1000 audio samples for each of the 24 instruments, yielding 24,000 total examples.

4.3 Feature Extraction

Each audio sample was processed using Octave to generate the feature set. The signal was first divided into equal width time windows. The number of time windows was selected to be twenty to yield 100-millisecond windows. Each of these 100-millisecond time windows was analyzed using a fast Fourier transform (FFT) to transform the data from the time domain into the frequency domain. This FFT transformation yielded an amplitude value, ranging [0, 1000] for each frequency point present in the analysis.

Frequency perception is a logarithmic concept but FFT analysis provides a resolution across a linear Hertz scale. Therefore, for example, the analysis provides a much lower resolution for the lowest note of the piano compared to the resolution

of the highest note. In order to group nearby frequencies into a single window, the vector was divided into ten exponentially increasing windows, where each frequency window is twice the size of the previous window, covering the range [0, 22050] Hertz. This generalization scheme allows the system to generalize over musical pitch.

Ten frequency windows were selected as a reasonable choice and will be empirically tuned in future work. For each of the ten frequency windows, the peak amplitude is extracted as the feature. The feature set for a single musical instrument example consists of ten frequency windows j for each of twenty time windows i, yielding 200 features per audio example. The feature extraction scheme is outlined in Figure 4.1. These 200 continuous features, ranging [0, 1000], are discretized into a variable number of bins using a supervised entropy-based binning scheme [29].

This feature set attempts to capture the unique and dynamic timbre of the each musical instrument by generalizing the changes in amplitude of groups of nearby partials over time for each instrument. Examples of the feature set for four musical instruments are visualized in Figures 4.2(a) - 4.2(d).

4.4 Models & Experimental Design

On this dataset, this project compared the performance of several Bayesian model structures in the task of musical instrument classification. The first model described is the naïve Bayes classifier. The remaining three Bayesian networks consist of variations of a grid-augmented naïve Bayes model, each adding different conditional dependencies in the time and frequency domains. For these descriptions, let f_j^i be the peak amplitude feature f at frequency window j for time window i, where $0 < i \leq 20$ and $0 < j \leq 10$.

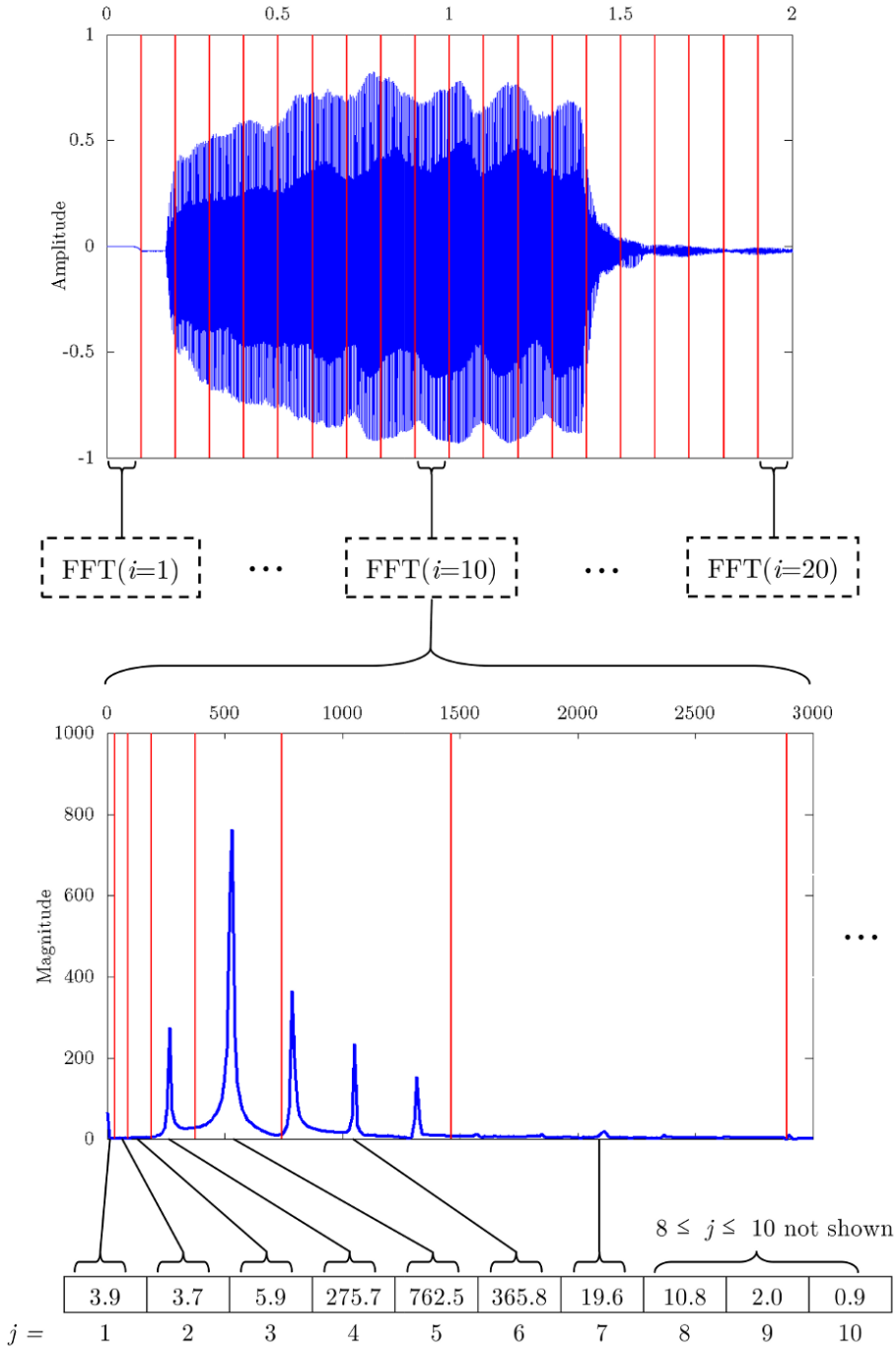
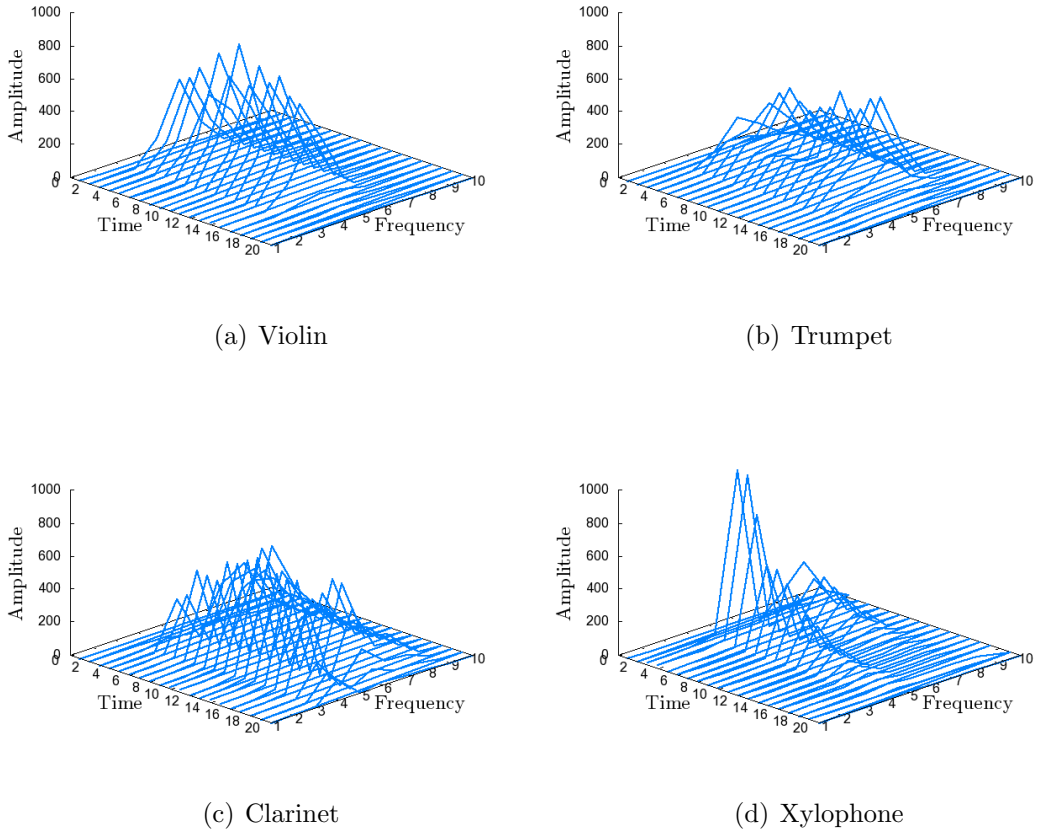


Figure 4.1: Each two second example is partitioned into twenty equal length windows. FFT analysis is performed on each 100 millisecond time window. The FFT analysis for i=10 is depicted. The FFT output is partitioned into ten exponentially increasing windows. For readability, only the first seven frequency windows are depicted above. The peak frequency from each window is extracted and used as a feature.



(a) Violin

(b) Trumpet

(c) Clarinet

(d) Xylophone

Figure 4.2: Visualization of the feature set for four different musical instruments each playing middle C at a mezzoforte dynamic level.

4.4.1 Naïve Bayes

For a baseline Bayesian model, we chose the common naïve Bayes classifier (NB). In the NB model, all evidence nodes are conditionally independent of each other, given the class. The formula for NB is shown as Equation 4.1 in which $P(c)$ is the class prior and $P(f \mid c)$ is the probability of a single feature within the feature set, given a particular class c. The NB network is shown graphically in Figure 4.3(a).

$$P(c \mid \mathbf{f}) = P(c) \times \prod_{f \in \mathbf{f}} P(f \mid c) \tag{4.1}$$

4.4.2 Frequency Dependencies

The second model is a Bayesian network with frequency dependencies (BN-F), in which each feature f_j^i is conditionally dependent on the previous frequency feature f_{j-1}^i within a single time window as shown in Figure 4.3(b), denoted as $f_{j-1}^i \rightarrow f_j^i$. Equation line 4.2a shows the class prior and the probability of the first row of the grid of features while line 4.2b defines the probability of the remaining features. There are no dependencies between the different time windows.

$$\begin{aligned} P(c \mid \mathbf{f}) &= P(c) \times \prod_{i=1}^{20} P(f_1^i \mid c) && \text{(4.2a)} \\ &\times \left(\prod_{i=1}^{20} \prod_{j=2}^{10} P(f_j^i \mid f_{j-1}^i, c) \right) && \text{(4.2b)} \end{aligned}$$

4.4.3 Time Dependencies

The third model, a Bayesian network with time dependencies (BN-T), contains conditional dependencies of the form $f_j^{i-1} \rightarrow f_j^i$ in the time domain, but contains no dependencies in the frequency domain (Figure 4.3(c)). Equation line 4.3a shows the class prior and the probability of the first column of the grid of features while line 4.3b defines the probability of the remaining features.

$$\begin{aligned} P(c \mid \mathbf{f}) &= P(c) \times \prod_{j=1}^{10} P(f_j^1 \mid c) && \text{(4.3a)} \\ &\times \left(\prod_{i=2}^{20} \prod_{j=1}^{10} P(f_j^i \mid f_j^{i-1}, c) \right) && \text{(4.3b)} \end{aligned}$$

4.4.4 Frequency and Time Dependencies

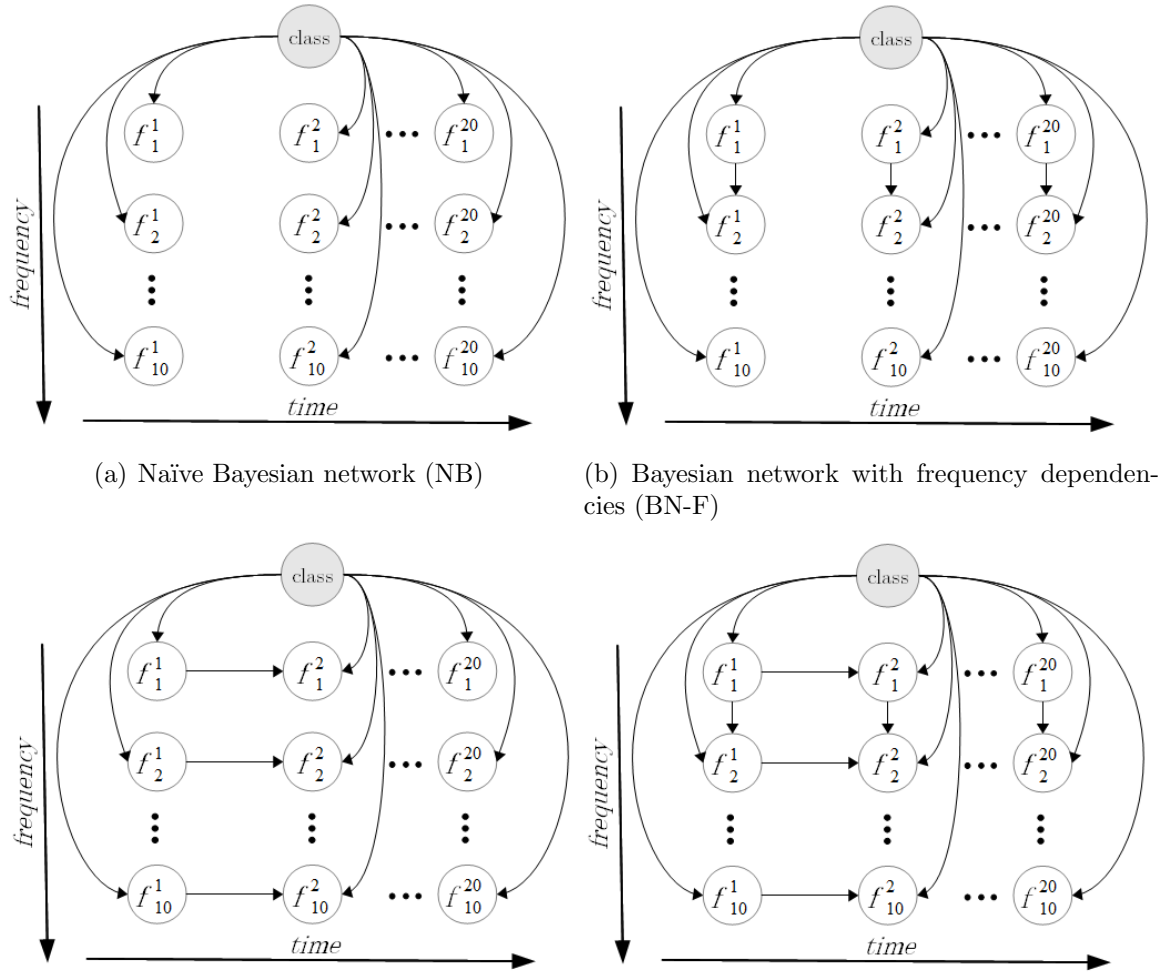
The final model, a Bayesian network with both time and frequency dependencies (BN-FT), is shown in Figure 4.3(d). The BN-FT model is a combination of BN-F and BN-T and contains dependencies of the form $f_j^{i-1} \rightarrow f_j^i$ and $f_{j-1}^i \rightarrow f_j^i$. Equation line 4.4a shows the class prior and the probability of the upper-leftmost node (f_1^1) of the feature grid. Line 4.4b shows the probability of first column of the grid, line 4.4c, that of the first row of the grid, and line 4.4d, that of the remaining features.

$$
\begin{aligned}
P(c \mid \mathbf{f}) &= P(c) \times P(f_1^1 \mid c) && \text{(4.4a)} \\
&\times \left(\prod_{i=2}^{20} P(f_1^i \mid f_1^{i-1}, c) \right) && \text{(4.4b)} \\
&\times \left(\prod_{j=2}^{10} P(f_j^1 \mid f_{j-1}^1, c) \right) && \text{(4.4c)} \\
&\times \left(\prod_{i=2}^{20} \prod_{j=2}^{10} P(f_j^i \mid f_j^{i-1}, f_{j-1}^i, c) \right) && \text{(4.4d)}
\end{aligned}
$$

4.4.5 Baseline Algorithms

To explore the advantages of time and frequency dependencies between features, the accuracies of the grid-augmented Bayesian models were compared with two support vector machines, a k-nearest neighbor classifier, and naïve Bayes. SVM and k-NN were chosen as the baseline algorithms for comparison to the Bayesian networks given the prevalence of these algorithms in the literature.

For the SVM, we selected both a linear (SVM-L) and polynomial kernel (see Equation 2.3) where $\delta = 2$ (SVM-Q). We also examined a radial basis function kernel and sigmoidal kernel; both scored at chance and were subsequently not included in



(a) Naïve Bayesian network (NB)

(b) Bayesian network with frequency dependencies (BN-F)

(c) Bayesian network with time dependencies (BN-T)

(d) Bayesian network with frequency and time dependencies (BN-FT)

Figure 4.3: Structure of the different Bayesian networks.

the experiments. For k-NN, we empirically examined values of k from 1 to 10. k-NN with $k = 1$ achieved the highest accuracy and was selected for use in all experiments.

4.4.6 Experimental Design

All experiments were run using ten-fold stratified cross-validation for training and testing. For the Bayesian networks, the parameter learning stage consisted of constructing the conditional probability tables (CPT) using counts from the training data. For all the Bayesian networks, the worst case size complexity of any variable's CPT is $O(n \cdot a^p)$ where $n = 200$ is the number of features, $9 \leq a \leq 42$ is the number of discretized states for any variable, and p is the maximum number of parents. For the most complex model, the BN-FT model, $p \leq 3$ for all variables.

In the testing phase, any event unseen in the training data results in a zero probability of the entire feature vector. To prevent this, we used the common technique of additive smoothing:

$$P(f_j^i) = \frac{x_i + \alpha}{N + \alpha \cdot d} \tag{4.5}$$

where $\frac{x_i}{N}$ is the probability of feature x_i, as indicated in the training data, and d is the total number of features [30]. The parameter α adds a small number of pseudo-examples to each possible feature value eliminating a possible count of zero that might result in a zero probability. A value of $\alpha = 0.5$ was used in all experiments.

4.5 Experiments & Results

4.5.1 Experiment 1: Instrument and Family Identification

The first experiment examined classification accuracy for both instrument identification ($n = 24$) and family identification ($n = 4$). The results are shown in Table 4.2.

The statistical significances using a paired student t-test with $p \leq 0.01$ are shown in Table 4.3.

All of the Bayesian networks, with the exception of naïve Bayes, outperformed both SVMs and k-NN. The model with frequency dependencies (BN-F) outperformed the model with time dependencies (BN-T). The combination of both frequency and time dependencies outperformed BN-F and BN-T in both tasks, more significantly so in the family identification task.

Table 4.2: Experiment 1 - Classification Accuracy (%) by instrument ($n = 24$) and by instrument family ($n = 4$)

Algorithm	Instrument	Family
NB	81.570	80.94
BN-F	97.525	92.87
BN-T	96.358	94.39
BN-FT	**98.252**	97.09
SVM-L	81.456	85.57
SVM-Q	93.55	95.65
k-NN	92.992	**97.31**

Table 4.3: Statistical significance of Experiment 1 using paired t-test with $p < 0.01$. Each cell indicates if the algorithm listed in the column performed significantly better (+), significantly worse (−), or not significantly different (0) when compared to the algorithm listed in the row. The first value is the significance of the instrument ($n = 24$) experiment and the second shows the family ($n = 4$) experiment.

Algorithm	NB	BN-F	BN-T	BN-FT	SVM-L	SVM-Q	k-NN
NB	—	+/+	+/+	+/+	0/+	+/+	+/+
BN-F	−/−	—	−/+	+/+	−/−	−/+	−/+
BN-T	−/−	+/−	—	+/+	−/−	−/+	−/+
BN-FT	−/−	−/−	−/−	—	−/−	−/−	−/0
SVM-L	0/−	+/+	+/+	+/+	—	+/+	+/+
SVM-Q	−/−	+/−	+/−	+/+	−/−	—	0/+
k-NN	−/−	+/−	+/−	+/0	−/−	0/−	—

Table 4.4: Confusion matrices for the family identification, showing classification counts. Bold values indicate a correct classification.

Algorithm	S	B	W	P	← classified as
NB	**4470**	21	327	162	String
	24	**3021**	944	11	Brass
	277	1923	**7799**	1	Woodwind
	220	320	324	**4134**	Percussion
BN-F	**4865**	15	107	13	String
	3	**3756**	239	2	Brass
	97	883	**9009**	111	Woodwind
	123	86	133	**4658**	Percussion
BN-T	**4921**	0	34	45	String
	13	**3612**	364	11	Brass
	173	600	**9223**	4	Woodwind
	27	55	21	**4897**	Percussion
BN-FT	**4923**	3	67	7	String
	1	**3627**	372	0	Brass
	19	198	**9783**	0	Woodwind
	4	15	13	**4968**	Percussion
SVN-L	**4692**	11	254	43	String
	47	**1265**	2685	3	Brass
	140	226	**9626**	8	Woodwind
	25	3	19	**4953**	Percussion
SVN-Q	**4670**	69	188	73	String
	84	**3667**	245	4	Brass
	119	190	**9680**	11	Woodwind
	42	5	14	**4939**	Percussion
k-NN	**4792**	56	107	45	String
	40	**3795**	162	3	Brass
	43	145	**9802**	10	Woodwind
	22	6	6	**4966**	Percussion

In many previous experiments, the family identification problem was found to be an easier problem than the instrument identification problem. Conversely, in this experiment, the Bayesian networks all performed less well on the family identification problem compared to the instrument identification problem. Both SVMs and k-NN,

however, both yielded improved classification accuracy on the family identification problem, consistent with the literature.

Confusion matrices for the family identification task are shown in Table 4.4. The Bayesian models showed increased confusion between brass and woodwind instruments compared to string or percussion instruments. The SVMs, k-NN and naïve Bayes, on the other hand, more often confused strings with either brass or woodwind compared to the Bayesian networks.

4.5.2 Experiment 2: Instrument Identification within Family

This experiment examines instrument classification by instrument family. Unlike Experiment 1, this experiment trains and tests only on instruments within the same family (Table 4.5). The dataset was divided into four separate datasets, one for each family, eliminating the possibility of confusion with instruments outside its own family. Ten-fold cross-validation is used on each of the family datasets.

Table 4.5: Experiment 2 - Classification accuracy (%) by instrument family

Algorithm	Strings	Woodwinds	Brass	Percussion
NB	89.76	84.58	92.43	99.64
BN-F	**99.86**	95.89	**99.70**	99.94
BN-T	99.12	95.56	99.36	99.92
BN-FT	99.60	**97.86**	99.58	**99.96**
SVM-L	98.66	92.01	98.65	98.18
SVM-Q	96.82	94.62	97.35	98.48
k-NN	98.72	92.67	98.63	99.72

Interestingly, the classification accuracy of strings, brass, and percussion exceeds 99% for all the Bayesian networks except naïve Bayes, whereas woodwinds, the largest set of instruments ($n = 10$), achieves 97.9% accuracy. For the strings, brass, and percussion, the BN-F and BN-FT achieves comparable accuracy, however, BN-FT

outperforms BN-F on the more difficult woodwind set. The percussion set achieve the highest accuracy for all algorithms, including the SVMs and k-NN.

4.5.3 Experiment 3: Accuracy by Dataset Size

This experiment examines the classification accuracy by instrument ($n = 24$), similar to Experiment 1, but as the dataset size varied from 100 to 1000 in increments of 100 for each instrument (Figure 4.4). The Bayesian network models converge to their respective optimal accuracy between 500 and 800 data samples per instrument. However, both the SVMs and k-NN continue to improve as the number of examples increase. It is possible that both would continue to improve accuracy if given more examples beyond 1000 examples per instrument. However, all the Bayesian models achieved much higher accuracy with far less examples than either SVMs or k-NN. This important result will be useful when extending this system to real-world examples extracted from commercial audio recordings.

4.6 Discussion

Many previous approaches, such as [8], reported the greatest difficulty with classifying string instruments over any other type of instrument. In our experiments, the Bayesian network models, however, had the greatest difficulty with woodwind instruments, although the Bayesian model still outperformed both SVMs and k-NN on the woodwind dataset. All algorithms tested performed extremely well on the percussion set, given the pronounced attack and immediate decay of these types of instruments, consistent with results from the literature.

The BN-FT model achieved comparable accuracy on both the instrument classification problem (n=24) and the family identification problem (n=4). However,

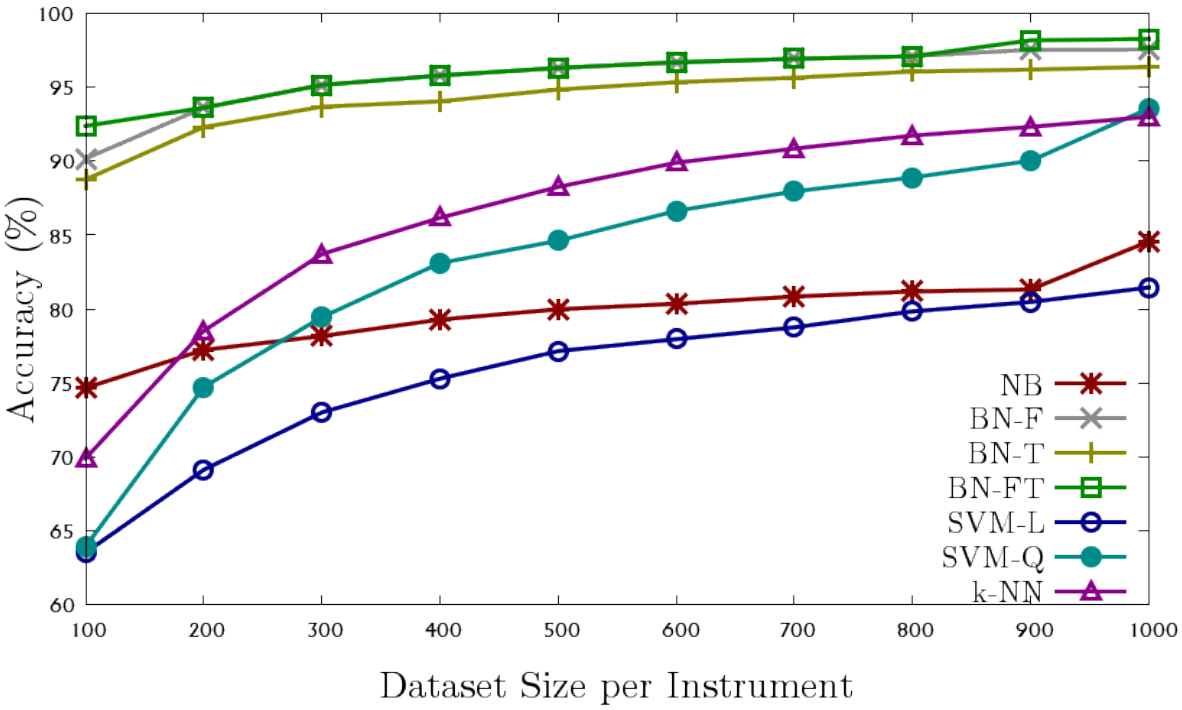


Figure 4.4: Experiment 3 - Accuracy (%) by number of examples per instrument for each model.

the BN-F and BN-T models each achieved better accuracy on individual instrument classification than they achieved on family identification. This result suggests that neither the frequency nor time dependencies themselves are sufficient to generalize across musical instrument families, but the combination of both sets of dependencies are needed. k-NN achieved much higher accuracy on the family identification problem compared to the instrument identification problem, unsurprisingly since k-NN is known not to scale well as the number of classes increases [31].

As shown in Table 4.4, the Bayesian models more often confused brass and woodwind instruments with each other compared to either string or percussion. This is perhaps unsurprising as our feature extraction scheme sought to capture the conditional relationships of changes in amplitude of frequencies over time. Woodwind and brass instruments are both classified as aerophones, instruments that generate sound by vibrating air, under the Hornbostel–Sachs system of scientific classification of musical instruments [32], suggesting that our feature extraction scheme may better model the physical acoustics of the instruments.

As the authors of [33] note, the choice of feature extraction scheme is crucial to the success of any music instrument classification system. Previous attempts to classify musical instruments have relied upon feature extraction schemes common in speech processing, most commonly the Mel-frequency cepstral coefficients (MFCC). Agostini *et al.* used a sparse set of nine spectral features to achieve 78.5% and 69.7% accuracy classifying 20 and 27 instruments, respectively, using an SVM [8]. Our feature extraction scheme, using 200 time and frequency varying features, achieved 93.6% accuracy classifying 24 instruments also using an SVM.

Although not directly comparable, these results imply that our feature extraction scheme better handles more instrument classes. While our system employs a considerably larger feature set, both feature extraction schemes are bounded by the

$O(n \log n)$ time complexity of the fast Fourier transform, where n is the number of samples in the audio file. Therefore we find no disadvantages in using a larger feature set. These results, when compared to the literature, also indicate that the feature extraction schemes that are optimized for speech recognition tasks may not be optimal in the musical instrument recognition task. Furthermore, these results also indicate that statistical dependencies modeling the changes in amplitude of partials over time, inspired by the human perception of timbre, are also useful in computational models.

4.7 Conclusion

In these preliminary results, we have presented a novel method for feature extraction, inspired by the psychoacoustic definition of timbre, that attempts to generalize the timbre of musical instruments probabilistically rather than rely on feature extraction schemes standard in speech recognition tasks. Furthermore, modeling conditional dependencies between both time and frequency (BN-FT) improves classification accuracy over either dependency individually (BN-F, BN-T) or none at all (NB).

The experiments presented here demonstrate that Bayesian networks are a valid approach to the classification of musical instruments. Overall, the BN-F, BN-T, and BN-FT models outperformed naïve Bayes, both SVMs, and k-NN. In addition to outperforming the SVMs and k-NN, the Bayesian models achieved desirable accuracy with far fewer examples and with less execution time, albeit with a larger feature space than other approaches in the literature. Given the success of our Bayesian approach in the single instrument classification problem, we will now modify our approach to attempt the more difficult multi-label classification problem.

CHAPTER 5
EXPERIMENTAL DESIGN

5.1 Overview

When multiple audio signals are mixed together, it is not possible to fully segregate them into the original discrete streams. This is known as the Cocktail Party Problem in the field of Cognitive Science and as the task of Auditory Scene Analysis in the field of Signal Processing.

This work proposes a Bayesian approach to multi-label classification of musical instruments. This approach attempts to determine the instruments present in a audio signal containing multiple instruments from a large set of instruments. The processes described in this section are fully extensible to the combination of an arbitrary number of instruments. In the experiments, however, I will begin with the task of determining pairs of instruments present in a recording and increase to three and eventually four instruments.

5.2 Dataset

A dataset similar to that discussed in Chapter 4 will be used in the dissertation experiments. The set of orchestral instruments described in 4.2 will be expanded to encompass the instruments available East-West Symphonic Orchestra library[1] and the Vienna Symphonic Library,[2] tentatively yielding 30 distinct instruments. 1000 examples of each instrument will again be used. Given the observation in Chapter

[1] http://www.soundsonline.com/Symphonic-Orchestra/

[2] http://www.vsl.co.at/en/211/1343/1344/950.vsl

4 that the presence of frequency dependencies improved accuracy over time dependencies alone, the length of each sample will be reduced down to one second from two. As in the previous experiments, the samples will be recorded at the **MON**tana **ST**udio for **E**lectronics and **R**hythm (MONSTER) at Montana State University at a 44.1k sampling rate, 16-bits per sample, and stored as a single channel waveform audio file (WAV).

These recordings of single instruments will be used to train the networks for the multi-label classification experiments. For the testing stage, examples of sound files containing two or more instruments are necessary. To generate files of two instruments sounding simultaneously, pairs of single instrument files will be mixed together. For instance, to cover all possible pairwise combinations, I will use 100 files for each possible pair of instruments. This results $\left(\frac{30!}{28!\cdot 2!}\right) * 100 = 43,500$ files. A similar process will be used to generate sets of three and four instruments.

Later experiments will include testing mixtures of more than two instruments, other datasets, such as the Musical Instrument portion of the commercial Real World Computing (RWV) Music Database[3] [34], the free University of Iowa Musical Instrument Samples collection,[4] and collections of real-world recordings.

5.3 Design

The proposed approach consists of three stages. The first stage is the Signature Matcher Bayesian network in which individual networks are created for each instrument that capture the probabilities of the ratios of the musical partials for a given fundamental at a given amplitude (Section 5.3.1). The second stage is the Feature

[3]http://staff.aist.go.jp/m.goto/RWC-MDB/

[4]http://theremin.music.uiowa.edu/MIS.html

Extractor (Section 5.3.2) in which the Signature Matcher Bayesian network will be queried, and features will be extracted from the FFT analysis of the example to be classified. In the third stage, the Instrument Classifier (Section 5.3.3), the system will determine if a particular instrument is present in the audio recording given the features extracted in the previous stage. Each of the stages are described below.

5.3.1 Signature Matcher

The Signature Matcher Bayesian network attempts to capture information about each harmonic partial relative to a known fundamental frequency. An individual Bayesian Network will be trained for each instrument. There will be a separate Bayesian model for each instrument and these networks will each be trained with examples containing only a single instrument, like the training data used in the preliminary experiments. These networks, once trained, will be used to attempt to extract features relevant to a single instrument from the spectral analysis of a signal containing multiple instruments.

To train these networks, an FFT will be run on an audio file containing a single recorded instrument. The peak harmonic amplitude within a single window that exceeds a minimum threshold will be extracted and its corresponding frequency saved. A windowing scheme will consist of overlapping windows each the size of a musical semitone. The lowest peak will be assumed to be the fundamental frequency f_0. Each significant peak above f_0 will be assumed to be a harmonic of the musical instrument and assigned to a partial number. The ratio of the frequency of this partial to the fundamental frequency will be extracted and used as evidence to train the Bayesian network.

A potential network structure is shown in Figure 5.1. f_0 indicates the fundamental frequency of the signal, and a_0, the amplitude of f_0. r_i indicates the ratio of partial i

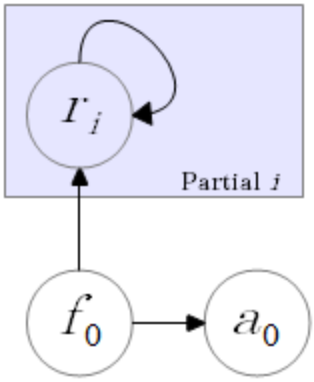


Figure 5.1: Potential network structure for the Signature Matcher Bayesian network.

relative to f_0. r_i is represented in a plate model that will repeat for partials $1 \leq i \leq 15$ with a dependency $r_i \rightarrow r_{i+1}$.

To give an example, consider a signal with peaks at 440, 885, and 1330 Hz. The lowest peak, 440 Hz, would be assumed to be the fundamental frequency and assigned to the variable f_0. The peaks of 885 and 1330 Hz would assumed to be the second and third partials, respectively. The ratios of these partials respective to the assumed fundamental would be 2.01 and 3.02 and assigned to variable r_2 and r_3 respectively. For explicit clarity, this simple example described integer and real-valued domains of the variables. In the experiments the values will be discretized into a variable number of bins using a supervised entropy-based binning scheme [29] as in the preliminary experiments.

5.3.2 Feature Extractor

The second stage of the system will be the Feature Extractor process. Any signal, once combined with another signal, cannot be segmented perfectly into the original signals. From the mixed signal, the original signals can be only estimated. The goal of this stage is to extract features from an audio stream containing multiple

instruments, separating them into discrete sets of features in which each represents a single instrument. This stage attempts to segment a complex spectral analysis into separate timbres using the template signatures learned in the previous section.

Given the FFT analysis of a file containing two instruments, this stage will consider each peak to be a fundamental frequency and extract partial information using the trained Signature Matcher network. More specifically, for each instrument, this stage will enumerate over all the peaks present in the analysis within the musical range of that instrument. Each peak will be considered to be a fundamental frequency and for this fundamental frequency the corresponding Signature Matcher network will be queried and a set of ratios returned. Within a window of a musical semitone, peaks at each of these ratios will be extracted and the amplitude of the peak extracted as a feature. These features will be used by the Instrument Classifier stage to determine if that instrument is present in the signal.

5.3.3 Instrument Classifier

Given the feature set extracted in the Feature Extraction stage, this stage will classify if each instrument is present in the musical signal for each potential fundamental. In accordance with the binary relevance approach to multi-label classification, a separate classifier will be trained for each instrument. Training will use the dataset of individual instruments. The testing stage, however, will use features from a multi-instrument mixture, extracted as described in the previous section.

A potential network structure is shown in Figure 5.2. f_0 indicates the fundamental frequency of the signal, a_0 designates the amplitude of f_0, r_i signifies the ratio of partial i relative to f_0, and a_i denotes the amplitude of partial i. r_i and a_i are represented in a plate model that will repeat for partials $1 \leq i \leq 15$ with dependencies $r_i \rightarrow r_{i+1}$ and $a_i \rightarrow a_{i+1}$. All nodes are connected to the class node $Instr$.

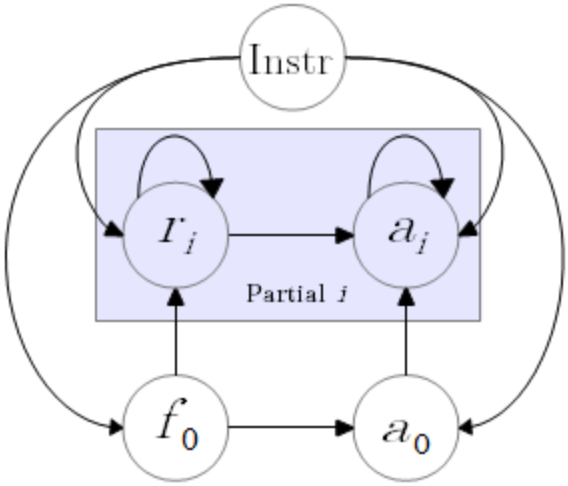


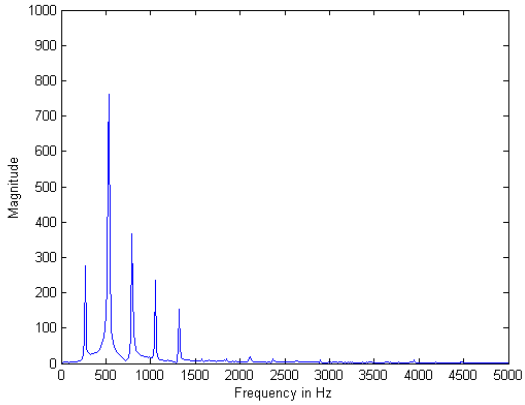
Figure 5.2: Potential network structure for the Instrument Classifier network.

This network structure assumes that all partials are dependent on the fundamental frequency. The amplitude a_i of any partial i is conditionally dependent on the ratio r_i as well as the amplitude of the fundamental a_0. The ratio r_i and the amplitude a_i of each partial i is dependent on the previous partial, inspired by the empirical success of the frequency dependencies described in the preliminary results.

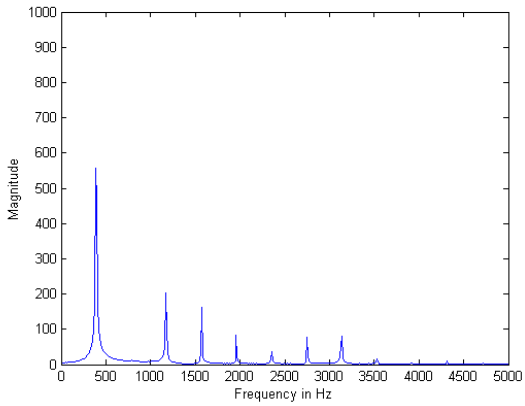
5.4 Example Walkthrough

Figure 5.3(a) shows the FFT analysis of a violin playing middle C, Figure 5.3(b), an oboe playing the G seven semitones (a perfect fifth) above the violin, and Figure 5.3(c) the mix of both both instruments playing together.

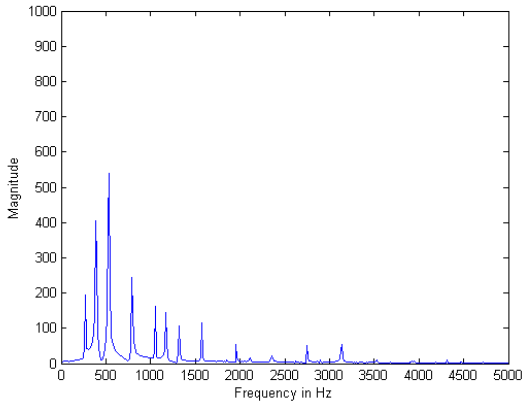
In the first stage, a Signature Matcher network will be trained for each instrument. This Signature Matcher network will learn for a particular network the probabilities of the ratios of the succeeding partials. In the second stage, the Feature Extractor will use the ratios learned in the Signature Matcher stage to extract partials from the FFT analysis.



(a) Violin playing C4, 261.6 Hz



(b) Oboe playing G4, 392.0 Hz



(c) Violin and Oboe mixed together

Figure 5.3: FFT of a Violin (a), Oboe (b), and Violin and Oboe playing a perfect fifth apart (c)

Consider Figure 5.3(c). If the system is attempting to classify if an oboe is present in the signal, the Feature Extractor must consider each peak in the analysis as a potential fundamental for that instrument. It would first begin with the first peak at 261 Hz. In this example, this is the fundamental frequency of the violin. The Feature Extractor would then extract partials according to the ratios learned by the Signature Matcher. These ratios as well as the amplitude of the partial found at that ratio will then be sent to the Instrument Classifier network. The Oboe Classifier network would ideally return false, as the amplitudes of the partials for that particular fundamental will be inconsistent with the training data for the oboe dataset.

The system would then return to the Feature Extractor stage and attempt the next possible fundamental, a peak at 392 Hz, which, in this example, is indeed the fundamental of the oboe. The process would repeat and if the Oboe Classifier network returns true, the system can terminate having determined that an Oboe is indeed present in the signal. The entire process would repeat for the Violin as well as all other instruments. The final classification will be the set of all instruments in which their respective Instrument Classifiers returned true.

5.5 Evaluation

As in the preliminary results described in Chapter 5, 10-fold cross-validation will be used in the experiments. To evaluate the accuracy of the system, the evaluation metric used in [35] will be employed. For example, for any prediction of a signal with two instruments present, there are three possible outcomes:

- If no instruments are matched correctly, assign a score of 0
- If only one of the two instruments is recognized correctly, assign a score of 0.5

- If both instruments are identified correctly, assign a score of 1.0.

The final score of the system will be the average of these score across all test instances, across all folds of the cross-validation. This evaluation scheme will be extended to accommodate groups or three and four instruments.

5.6 Contributions

This proposal outlines a new approach for the multi-label classification of musical instrument timbre. As discussed in Section 3.3, multi-label classification of musical instrument timbre is a difficult problem and most systems cannot adequately handle realistic datasets (e.g., live processing or commercial recordings) with an acceptable level of accuracy. The system proposed here trains on single music instruments, segments the timbre of a multi-instrument timbre using signature matching, and classifies using a series of binary relevance classifiers.

In addition to the contributions to Music Information Retrieval and multi-label classification, this system proposes a novel feature extraction scheme. Although other systems have attempted variations of template matching within spectra [20, 22, 27], my approach is the first to propose training graphical models to allow for probabilistic template matching of musical spectra. This approach, I hypothesize, will generalize much better and achieve better performance on noisy or real-world data compared to other systems in literature. Furthermore, because my Template Matching Bayesian network captures the ratio of partials to the fundamental, my system, unlike most others, is designed to handle both harmonic and inharmonic instruments.

My approach has numerous benefits and advantages. First and foremost, I predict my system will scale well to many instruments and additional instruments can be added later merely by training a Signature Matching Bayesian network for that

instrument. Secondly, while training the networks may take significant time and computational resources, the trained Bayesian networks can be stored and retrieved for use later. This is a distinct advantage over multi-label k-NN approaches.

In the testing stage, the complexity of the feature extraction is bounded by the $O(n \log n)$ time complexity of the fast Fourier transform, where n is the number of samples in the audio file. This allows for quick classification, given trained networks. While these features must be classified by a separate binary classifier for each instrument, the system of binary relevance classifiers readily lends itself to parallelization. Such a system would, for example, allow for an efficient offline labeling of a large scale dataset of musical recordings, or perhaps a plugin for a program such as VLC that analyzes the instruments present in a signal in real-time.

5.7 Work Plan

The following section outlines a work schedule for progress on the dissertation.

1. **Dataset Generation**

   As described in Section 5.2, I will create a new dataset and use several existing datasets.

   (a) **Single Instrument:** This dataset will consist 1000 one-second examples of 30 different, individual musical instruments. This dataset will be used to train the Signature Matcher Bayesian networks.

   (b) **Multi-Instrument Mixtures:** This dataset will consists of mixtures of two, three, and four instruments derived from random mixes of instruments in the Single Instrument dataset described above, including examples of more than one instance of the same instrument playing simultaneously.

(c) **Existing Datasets:** For the testing phase, examples from the Real World Computing (RWV) Music Database and the University of Iowa Musical Instrument Samples will be used. Like my Single Instrument dataset, these are collections of recordings of single instruments. I will randomly derive mixtures of instruments in a manner similar to my Multi-Instrument Mixtures described above. Use of these datasets will make my approach more comparable to other studies that have used these datasets.

(d) **Real-world Recordings:** Also for the testing phase, I will find sources of real-world recordings. One potential source is Norton's multi-volume CD anthology of examples from Chamber and Orchestral Music,[5] commonly used as reference in music history courses.

2. **Algorithm Design**

These Bayesian networks will be designed from scratch using the network structures described in Section 5.3 and shown in Figures 5.1 and 5.2. The Feature Extractor algorithm will also be designed by hand but will use an existing implementation of a Fast Fourier Transform (FFT) implementation from a numerical analysis toolkit.

3. **Evaluation**

My approach to multi-label classification of musical instrument timbre will be empirically evaluated through a number of experiments. This system will be trained on single instrument examples but tested on multi-instrument mixtures of two, three, and four examples.

[5]http://www.wwnorton.com/college/music/grout7/home.htm

(a) **Training:** Training of these networks will derive from the example of single musical instruments. The features will be extracted as described in Section 5.3.1 and the values discretized. The values of the variables will derive from the counts of these discretized bins in the training data.

(b) **Testing:** Testing of these networks will use features extracted from the multi-instrument signals as described in Section 5.3.2. Experiments will be performed testing on two, three, and four instrument mixtures.

(c) **Comparison:** The results of my approach will be empirically compared to several existing approaches for multi-label classification. In addition to common algorithms such as SVM, I will consider the following common multi-label classification algorithms:

- the Binary Relevance k-NN (BRkNN) algorithm and the Multi-label k-NN (MLkNN) [36],
- Random k-labelsets (Ra$k$el) [37], and
- Back-Propagation Multi-Label Learning (BPMLL) from the MuLaN toolkit [38].

4. **Extensions**

The above steps outline the primary goals of this research. Given the completion of these goals, I will explore and incorporate into the dissertation these extensions:

(a) **Structure Learning:** Figures 5.1 and 5.2 show two potential network structures for the Signature Matcher and Instrument Classifier networks, respectively. These network structures have been determined by hand given reasonable assumptions using domain knowledge about the harmonic

series and musical instrument timbre. I will begin with these networks for the initial experiments. In later experiments, I will use structure learning to determine an ideal network structure for this problem given our dataset. It is quite possible that different structures may be learned for each different instrument.

(b) **Latent Variables:** A latent variable is a hidden variable whose values are not directly observable but rather inferred. Upon completion of the above goals, I will explore incorporating latent variables into the network structures of Figures 5.1 and 5.2. Training a network with latent variables requires the use of inference and I will use a common inference engine, such as Smile.[6]

(c) **Spectral Clustering:** Spectral clustering is a technique for dimensionality reduction and feature extraction which uses the spectrum of the similarity matrix of a graph representation of the data to cluster in fewer dimensions [39]. As an alternative approach to my above-mentioned feature extraction, I will explore spectral clustering as a mean to reduce the dimensionality of my feature set.

(d) **Domains:** In this proposal, the feature extraction scheme is described for the segmentation of musical timbre. With modification to the structures of the Bayesian networks, this scheme could be extended to a number of other spectral domains, such as radar, sonar, anomaly detection, speaker identification, or elimination of noise in signals. Given adequate time, I will run experiments testing my approach on one or more other domains outside of the musical instrument classification problem.

[6] http://genie.sis.pitt.edu/download/software.html

REFERENCES CITED

[1] P.J. Donnelly, C.J. Limb. *Encyclopedia of Neuroscience*, Chapter Music, pages 1151–1158. Elsevier, 2009.

[2] T. Cover, P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[3] B.E. Boser, I.M. Guyon, V.N. Vapnik. A training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152. ACM, 1992.

[4] N. Friedman, D. Geiger, M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2):131–163, 1997.

[5] J.M. Grey. Multidimensional Perceptual Scaling of Musical Timbres. *Journal of the Acoustical Society of America*, 61(5):1270–1277, 1977.

[6] I. Fujinaga, K. MacMillan. Realtime Recognition of Orchestral Instruments. *Proceedings of the International Computer Music Conference*, Volume 141, page 143, 2000.

[7] J. Marques, P.J. Moreno. A Study of Musical Instrument Classification Using Gaussian Mixture Models and Support Vector Machines. *Cambridge Research Laboratory Technical Report Series*, 4, 1999.

[8] G. Agostini, M. Longari, E. Pollastri. Musical instrument timbre classification with spectral features. *EURASIP Journal on Applied Signal Processing*, 2003:5–14, 2003.

[9] I. Kaminskyj, T. Czaszejko. Automatic Recognition of Isolated Monophonic Musical Instrument Sounds Using kNNC. *Journal of Intelligent Information Systems*, 24(2):199–221, 2005.

[10] B. Kostek. Musical Instrument Classification and Duet Analysis Employing Music Information Retrieval Techniques. *Proceedings of the IEEE*, 92(4):712–729, 2004.

[11] A. Wieczorkowska. Classification of Musical Instrument Sounds Using Decision Trees. *Proceedings of the 8th International Symposium on Sound Engineering and Mastering, ISSEM*, Volume 99, pages 225–230, 1999.

[12] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1997.

[13] A. Eronen. Musical Instrument Recognition Using ICA-based Transform of Features and Discriminatively Trained HMMs. *Proceedings of the Seventh International Symposium on Signal Processing and Its Applications*, Volume 2, pages 133–136. IEEE, 2003.

[14] H. Zhou, T. Hermans, A.V. Karandikar, J.M. Rehg. Movie genre classification via scene categorization. *Proceedings of the international conference on Multimedia*, pages 747–750. ACM, 2010.

[15] S. Gao, W. Wu, C.H. Lee, T.S. Chua. A mfom learning approach to robust multiclass multi-label text categorization. *Proceedings of the twenty-first international conference on Machine learning*, page 42. ACM, 2004.

[16] M.R. Boutell, J. Luo, X. Shen, C.M. Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.

[17] K. Trohidis, G. Tsoumakas, G. Kalliris, I. Vlahavas. Multilabel classification of music into emotions. *Proc. 9th International Conference on Music Information Retrieval (ISMIR 2008), Philadelphia, PA, USA*, Volume 2008, 2008.

[18] G. Tsoumakas, I. Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.

[19] J. Read, B. Pfahringer, G. Holmes, E. Frank. Classifier chains for multi-label classification. *Machine Learning and Knowledge Discovery in Databases*, pages 254–269, 2009.

[20] J.G.A. Barbedo, G. Tzanetakis. Musical instrument classification using individual partials. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(1):111–122, 2011.

[21] S. Essid, G. Richard, B. David. Instrument recognition in polyphonic music based on automatic taxonomies. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(1):68–80, 2006.

[22] J.J. Burred, A. Robel, T. Sikora. Polyphonic musical instrument recognition based on a dynamic model of the spectral envelope. *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 173–176. IEEE, 2009.

[23] P. Somerville, A.L. Uitdenbogerd. Multitimbral musical instrument classification. *International Symposium on Computer Science and its Applications*, pages 269–274. IEEE, 2008.

[24] J. Eggink, G.J. Brown. Application of missing feature theory to the recognition of musical instruments in polyphonic audio. *Proc. ISMIR*, pages 125–131, 2003.

[25] P. Jincahitra. Polyphonic instrument identification using independent subspace analysis. *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, Volume 2, pages 1211–1214. IEEE, 2004.

[26] P. Hamel, S. Wood, D. Eck. Automatic identification of instrument classes in polyphonic and poly-instrument audio. *Proc. Int. Conf. Music Information Retrieval*, 2009.

[27] T. Kitahara, M. Goto, K. Komatani, T. Ogata, H.G. Okuno. Instrument identification in polyphonic music: Feature weighting to minimize influence of sound overlaps. *EURASIP Journal on Applied Signal Processing*, 2007(1):155–155, 2007.

[28] P. Donnelly, J. Sheppard. Classification of musical timbre using bayesian networks. *Computer Music Journal*. Submitted for publication.

[29] U.M. Fayyad, K.B. Irani. On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, 8(1):87–102, 1992.

[30] S.F. Chen, J. Goodman. An empirical study of smoothing techniques for language modeling. *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, pages 310–318. Association for Computational Linguistics, 1996.

[31] P. Jain, A. Kapoor. Active learning for large multi-class problems. *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 762–769. IEEE, 2009.

[32] E.M. von Hornbostel, C. Sachs. *Systematik der Musikinstrumente*. Behrend, 1914.

[33] J.D. Deng, C. Simmermacher, S. Cranefield. A study on feature analysis for musical instrument classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 38(2):429–438, 2008.

[34] M. Goto, i in. Development of the rwc music database. *Proceedings of the 18th International Congress on Acoustics (ICA 2004)*, Volume 1, pages 553–556, 2004.

[35] E. Spyromitros Xioufis, G. Tsoumakas, I. Vlahavas. Multi-label learning approaches for music instrument recognition. *Foundations of Intelligent Systems*, pages 734–743, 2011.

[36] E. Spyromitros, G. Tsoumakas, I. Vlahavas. An empirical study of lazy multilabel classification algorithms. *Artificial Intelligence: Theories, Models and Applications*, pages 401–406, 2008.

[37] G. Tsoumakas, I. Katakis, I. Vlahavas. Mining multi-label data. *Data Mining and Knowledge Discovery Handbook*, pages 667–685, 2010.

[38] G. Tsoumakas, J. Vilcek, E. Spyromitros, I. Vlahavas. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 1:1–48, 2010.

[39] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.