Patrick J. Donnelly*[†] and John W. Sheppard[†]
*School of Music
Montana State University
P.O. Box 173420
Bozeman, Montana 59715, USA
[†]Department of Computer Science
Montana State University
P.O. Box 173880
Bozeman, Montana 59715, USA
{patrick.donnelly2,john.sheppard}@cs.montana.edu

Classification of Musical Timbre Using Bayesian Networks

Abstract: In this article, we explore the use of Bayesian networks for identifying the timbre of musical instruments. Peak spectral amplitude in ten frequency windows is extracted for each of 20 time windows to be used as features. Over a large data set of 24,000 audio examples covering the full musical range of 24 different common orchestral instruments, four different Bayesian network structures, including naive Bayes, are examined and compared with two support vector machines and a *k*-nearest neighbor classifier. Classification accuracy is examined by instrument, instrument family, and data set size. Bayesian networks with conditional dependencies in the time and frequency dimensions achieved 98 percent accuracy in the instrument classification task and 97 percent accuracy in the instrument family identification task. These results demonstrate a significant improvement over the previous approaches in the literature on this data set. Additionally, we tested our Bayesian approach on the widely used Iowa musical instrument data set, with similar results.

The identification of musical instruments in audio recordings is a frequently explored, yet unsolved, machine learning problem. 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 ability of a computer to learn to identify musical instruments is an important problem within the field of music information retrieval, with high commercial value. For instance, companies could automatically index their music libraries based on the musical instruments present in the recording, allowing search and retrieval by specific musical instrument. Timbre identification is also important to the tasks of musical genre categorization, automatic score creation, and track separation.

This work investigates classification of single, monophonic musical instruments using several different Bayesian network structures and a feature-extraction scheme based on a psychoacoustic definition of timbre. The results of this seminal use of graphical models in the task of musical instrument classification are compared with the baseline algorithms of support vector machines and a *k*-nearest neighbor classifier.

Computer Music Journal, 37:4, pp. 70–86, Winter 2014
doi:10.1162/COMJ_a_00210

Timbre

When a musical instrument plays a note, we perceive a musical pitch, the instrument playing that note, and other aspects, like loudness. Timbre, or tone color, is the psychoacoustic property of sound that allows the human brain to readily distinguish between two instances of the same note, each played on a different instruments. The primary musical pitch we perceive is usually the first harmonic partial, known as the fundamental frequency. Pitched instruments are those whose partials are approximate integer multiples of the fundamental frequency. With the exception of unpitched percussion, orchestral instruments are pitched. The perception of timbre depends on the presence of harmonics (i.e., spectrum), as well as the fine timing (envelope) of each harmonic constituent (partial) of the musical signal (Donnelly and Limb 2009).

Algorithms

This work compares three types of algorithms on the machine learning task of timbre classification. This section briefly explains each of the algorithms we used.

Nearest Neighbor

The *k*-nearest neighbor (*k*-NN) 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 a multidimensional feature space. Formally, let $\mathcal{X}$ be a space of points where each point $x_i \in \mathcal{X}$ is defined as $x_i = \langle \{x_i^1, \ldots, x_i^d\}; c_i \rangle$ and $\mathcal{X}_{tr} \subset \mathcal{X}$ be a set of training examples. For $x_q \in \mathcal{X} - \mathcal{X}_{tr}$ find $r \in \mathcal{X}_{tr}$ such that $\forall x \in \mathcal{X}_{tr},\ x \neq r,\ D(x_q, r) < D(x_q, x)$ and return the associated class label c_r (Cover and Hart 1967). In other words, each query example f_q in the test set will be compared to a subset of examples from the training set, using a distance metric, and the most common class label among these k neighbors will be assigned to f_q.

Support Vector Machine

The support vector machine (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 a discriminant-based method for classification or regression, following the approach of Vapnik (1999). The SVM algorithm constructs a hyperplane in high dimensional space that represents the largest margin separating two classes of data. The SVM is defined as the hyperplane $\mathbf{w}^\top \cdot \Phi(\mathbf{f}) - b = 0$ that solves the following quadratic programming problem:

$$\text{minimize} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + C \cdot \sum_i \xi_i \right\} \tag{1}$$

subject to:

$$y(\mathbf{w}^\top \cdot \Phi(\mathbf{f}) - b) \geq 1 - \xi_i,\ \xi_i \geq 0 \tag{2}$$

where
$\mathbf{f}$ is a vector of features,
$\mathbf{w}^\top$ is the discriminant vector,
C is a regularizing coefficient,
ξ_i is a slack variable,
b is the bias offset,
y is the class label such that $y \in \{-1, +1\}$,
and the kernel function $K(\mathbf{f_i}, \mathbf{f_j}) = \Phi(\mathbf{f_i})^\top \cdot \Phi(\mathbf{f_j})$ is the inner 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 3), the features are projected into a higher-order space. This allows the algorithm to fit the maximum margin hyperplane in the transformed feature space, which is no longer linear in the original space (Boser, Guyon, and Vapnik 1992).

$$K(\mathbf{f_i}, \mathbf{f_j}) = (\mathbf{f_i} \cdot \mathbf{f_j})^\delta \tag{3}$$

Bayesian Networks

Bayesian networks are probabilistic graphical models that are composed 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 the node's parent variables. The Bayesian classifier without latent variables is defined as:

$$\text{classify}(\mathbf{f}) = \underset{c \in C}{\text{argmax}}\, P(c) \prod_{f \in \mathbf{f}} P(f \mid \text{parent}(f)) \tag{4}$$

where $P(c)$ is the prior probability of class c and $P(f \mid \text{parent}(f))$ is the conditional probability of feature f given the values of the variable's parents. The classifier finds the class label which has the highest probability of explaining the values of the feature vector (Friedman, Geiger, and Goldszmidt 1997).

Previous Work

Beginning with initial investigations of psychoacoustician John Grey (1977), the task of musical instrument identification has relied on clustering techniques. Fujinaga and MacMillan (2000) created

Table 1. Comparison of Three Approaches to Instrument Identification

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**

Agostini, Longari, and Pollastri (2003) compared a support vector machine (SVM), k-nearest neighbor (k-NN) classifier, and quadratic discriminant analysis (QDA) in the task of identification of musical instruments. The authors compared three different sets of instruments of varying size as well as the identification of the instrumental family (strings, woodwinds, or brass) and percent accuracy is listed. **Boldface** values indicate the approach with the highest accuracy for each experiment.

a *k*-NN system that achieved 68 percent instrument classification on a large database of 23 different recorded instruments. Kaminskyj and Czaszejko (2005) used *k*-NN to achieve 93 percent instrument classification and 97 percent instrument family recognition on a set of 19 instruments.

In the 2000s, investigators began to explore other techniques. In a seminal study using SVM, Marques and Moreno (1999) classified 200 msec of recorded audio for eight musical instruments, using 16 Mel-frequency cepstral coefficients as features. The authors achieved 70 percent accuracy using a "one-versus-all" multi-class SVM with a polynomial kernel, which outperformed the 63 percent accuracy using Gaussian mixture models (GMMs). On a set of 10 instruments, Essid, Richard, and David (2006) achieved 87 percent accuracy using an SVM with a radial basis function kernel, outperforming the 82 percent accuracy of a GMM.

In 2003, a study demonstrated the ability of SVMs to outperform *k*-NN on the task of musical instrument identification. Agostini, Longari, and Pollastri (2003) used a set of nine spectral features and compared the results of an SVM, *k*-NN, and quadratic discriminant analysis. Their results on three different sets of instruments are shown in Table 1. For the 27-instrument set, those authors also tested instrument family discrimination (i.e., strings versus woodwinds versus brass) and achieved 80.8 percent accuracy using an SVM, compared with 76.2 percent using *k*-NN. A recent study explored the efficacy of the SVM on the family identification task for a data set that included non-Western instruments. Liu and Xie (2010) achieved 87 percent accuracy on a set of eight instrument families covering both Western and Chinese instruments.

Although *k*-NN and SVM remain the most commonly used system for timbre classification, a few other approaches have been utilized. Kostek (2004) used a multilayer feedforward neural network to identify twelve musical instruments playing a wide variety of articulations using a combination of MPEG-7 and wavelet-based features. She achieved 71 percent accuracy, ranging from 55 percent correct identification of the English horn to 99 percent correct identification of the piano. 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 (Wieczorkowska 1999) used a binary decision tree, a variation of the C4.5 algorithm (Quinlan 1993), to classify 18 instruments using 62 features, yielding 68 percent classification accuracy. On a limited set of 6 instruments, Benetos, Kotti, and Kotropoulos (2006) achieved 95 percent accuracy using a non-negative matrix factorization classifier with MPEG-7 spectral features.

Several recent studies have explored the utility of temporal information in instrument classification tasks. Through a variety of experiments, Joder, Essid, and Richard (2009) determined that including temporal information can significantly improve classification performance, albeit at the cost of the problem dimensionality. Using an SVM, the authors achieved 84 percent accuracy on a set of eight instruments, outperforming both the GMM and hidden Markov model (HMM) classifiers. Burred, Robel, and Sikora (2010) used Principal Component Analysis to create prototypes of the temporal evolution of timbral envelopes. The authors used non-stationary Gaussian processes to classify a set of five instruments with 95 percent accuracy.

Recently, investigators have begun exploring instrument identification in the presence of polyphonic mixtures of instruments. Using a GMM, Heittola, Klapuri, and Virtanen (2009) achieved 59 percent accuracy on six-note polyphonic mixtures, chosen from a set of 19 instruments. Burred, Robel, and Sikora (2009) examined Euclidean distances between spectral envelope shapes, achieving accuracies of 77 percent on two instrument mixtures, 43 percent accuracy on three instrument mixtures, and 40 percent on four instrument mixtures. Barbedo and Tzanetakis (2011) used a novel instrument classification method of identifying isolated partials in a signal. Over a large set of 25 instruments and using a linear SVM, the authors achieved 62 percent accuracy in identifying instruments from mixtures of two to five instruments.

Despite a few attempts using other learning strategies, the focus in the literature remains dominated by SVM and *k*-NN for this task. Although Bayesian networks, most commonly the HMM, have been widely used in the field of natural language processing for speech recognition, phoneme identification, and other tasks (Jelinek 1997), Bayesian networks have not been widely used for the problem of musical instrument identification. Eronen (2003) presented preliminary results using a continuous-density HMM to classify seven different instrument groupings, but not individual instruments, achieving accuracies between 45 percent and 64 percent.

Data Generation and Feature Extraction

Feature extraction is a form of dimensionality reduction in which, for the purposes of this task, the audio files are transformed to a small vector of highly relevant numeric features. Recently, attention in the literature has centered on the task of feature identification (for a review, see Deng, Simmermacher, and Cranefield 2008) rather than on the choice of learning strategy. In addition to differing data sets, most of the studies in the literature have used varied sets of features, rendering any direct comparison of studies in the literature impossible. In order to compare our Bayesian approach to timbre classification against the methods commonly used in the literature, we created a data set, defined a spectral-based feature-extraction scheme, and empirically compared our Bayesian classifiers to a *k*-NN and two SVM classifiers. Additionally, we tested our feature-extraction scheme and our classifiers on the commonly used Iowa data set.

Table 2. EastWest Data Set of Instruments

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

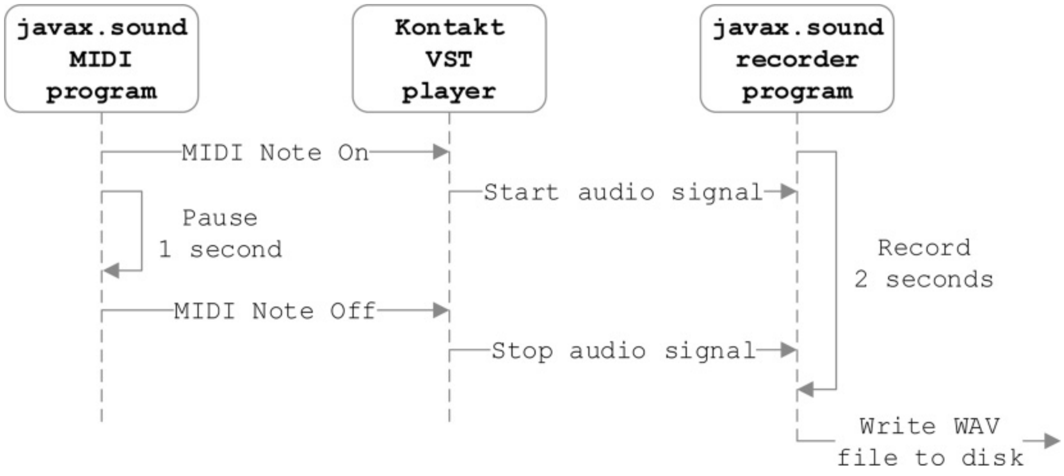
The 24 instruments in the data set grouped into instrument families. The bottom row indicates the number of instruments in each family.

EastWest Data Set

For our experiments, we create a data set (EastWest) that contains 1,000 audio examples for each musical instrument, covering 24 different orchestral instruments (see Table 2). Each audio file is 2 sec in duration, consisting of the instrument sustaining a single note for 1 sec, 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 (EastWest 2013) at the Montana Studio for Electronics and Rhythm (MONSTER) at Montana State University.

Figure 1 shows an overview of the data generation process. For each musical instrument, Native Instruments' Kontakt Virtual Studio Technology (VST) player (Native Instruments 2013) loads the respective samples from the EastWest sample library. For each musical example, a MIDI control sequence is sent from a Java program to the Kontakt sampler for rendering to audio. The interaction between Java and the VST player is handled by the jVSTwRapper interface (jVSTwRapper 2013). Using the EastWest Symphonic Library, the VST player produces an audio signal that corresponds to the parameters of the MIDI message. The resulting

Figure 1. For a selected pitch and dynamic level, MIDI control signals are transmitted to the Kontakt VST player. The VST player renders an audio signal corresponding to the parameters of the MIDI messages. This signal is then recorded by another Java program and the resulting sample is saved to disc as a WAV file.



audio stream is recorded in another Java program using the javax.sound package. The samples are recorded at a 44.1kHz sampling rate, 16 bits per sample, and stored as a single channel waveform audio file (WAV).

The pitch is randomly sampled uniformly covering the entire musical range of the instrument. The dynamic level is also sampled uniformly of the MIDI velocity parameter in the range [40, 105], covering the dynamic range pianissimo to fortissimo. Although pitch and dynamic values could each be repeated, very few duplicates of the pitch and dynamic pairing occurred for any single instrument. In total, there are 1,000 audio samples for each of the 24 instruments, yielding 24,000 total examples.

The data set is then normalized to the range [0, 1] using the audio utility *normalize* (Vaill 2013). The files are batch normalized to scale the loudest gain in any of the files to a value of one and adjusting all the other files by this offset. This method preserves the relative dynamic levels between example files. The data set is available online at nisl.cs.montana.edu/instruments.

Iowa Data Set

The Iowa data set, created by the Electronic Music Studios at the University of Iowa, contains scales of 21 different musical instruments each at three different dynamic levels: pianissimo, mezzoforte, and fortissimo (University of Iowa 2013). We separated these scales into individual files each containing a single note using the *Sound eXchange* (SoX) audio program (Bagwall et al. 2013). For the purposes of these experiments, the bowed and *pizzicati* samples of the Violin, Viola, Cello, and Contrabass are considered to be eight separate classes. This data set contains 4,521 samples covering 25 different instrument classes (see Table 3). The number of samples for each instrument varies, ranging from 70 examples of the Bass Trombone

Table 3. Iowa Data Set of Instruments

Strings	*Woodwinds*	*Brass*
Piano	Alto Flute	
Guitar	Flute	
Violin	Bass Flute	French Horn
Viola	Soprano Saxophone	Trumpet
Cello	Alto Saxophone	Trombone
Bass	Bb Clarinet	Bass Trombone
Violin Pizzicato	Eb Clarinet	Tuba
Viola Pizzicato	Bass Clarinet	
Cello Pizzicato	Oboe	
Bass Pizzicato	Bassoon	
10	10	4

The 25 instruments in the data set grouped into instrument families. The bottom row indicates the number of instruments in each family.

up to 352 examples of the Cello. The samples are remixed in mono, 44.1 kHz, 16 bit, clipped to 2 sec in duration, and batch normalized to the unit range [0,1] using the the normalization strategy described in the previous section.

Feature Extraction

Each audio sample is processed in MATLAB using the "fastest Fourier transform in the West" (FFTW) algorithm. The signal is first divided into time windows of equal width. The number of time windows is selected to be 20, to yield 100-msec windows. Each of these 100-msec time windows is analyzed using a fast Fourier transform (FFT) to transform the data from the time domain into the frequency domain. This FFT transformation yields an amplitude value for each frequency point present in the analysis. These amplitude values range from 0 to 1,000, as specified by the default settings of the FFTW algorithm.

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 with the resolution of the highest note. In order to group nearby frequencies into a single window, the vector is divided into ten exponentially increasing windows, where each frequency window is twice the size of the previous window, covering the range from 0 to 22,050 Hz. This scheme allows the system to work equally well across the frequency range.

The choice of ten as the number of frequency windows was reasonable 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 20 time windows i, yielding 200 features per audio example. The feature-extraction scheme is outlined in Figure 2.

These 200 continuous features, in the range $[0, 1000]$, are discretized into a variable number of bins using a supervised entropy-based binning scheme (Fayyad and Irani 1992). Entropy provides a measure of purity of a certain interval. Let k correspond to the number of class labels and p_{ij} correspond to the conditional probability of class j occuring in the ith interval. The entropy h_i of the interval i is given by the equation:

$$h_i = -\sum_{i=1}^{k} p_{ij} \log p_{ij} \tag{5}$$

The total entropy H of the discretization is the weighted average of the individual entropies:

$$H = \sum_{i=1}^{n} w_i h_i \tag{6}$$

where m is the number of values in the data set, $w_i = m_i/m$ is the fraction of the values in the ith interval, and n is the number of intervals.

Entropy-based discretization considers all possible bisections of an interval, computes the associated entropies, and retains the bisection with the lowest entropy. The process continues by selecting the next interval with the highest entropy and repeating the process until the stopping criterion, given by Fayyad and Irani (1993), is reached.

This feature set attempts to capture the unique and dynamic timbre of 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 Figure 3.

Models and Experimental Design

This project compares the performance of several Bayesian model structures in the task of musical instrument classification on these data sets. The first model described is the naive Bayes classifier. The remaining three Bayesian networks consist of variations of a grid-augmented naive Bayes model, each adding different conditional dependencies in the time and frequency domains. For all of 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 2. Feature extraction. Each 2-sec example is partitioned into 20 windows of equal length. FFT analysis is performed on each 100-msec 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 here. The amplitude of the frequency peak with the highest magnitude within each window is extracted and used as a feature.*

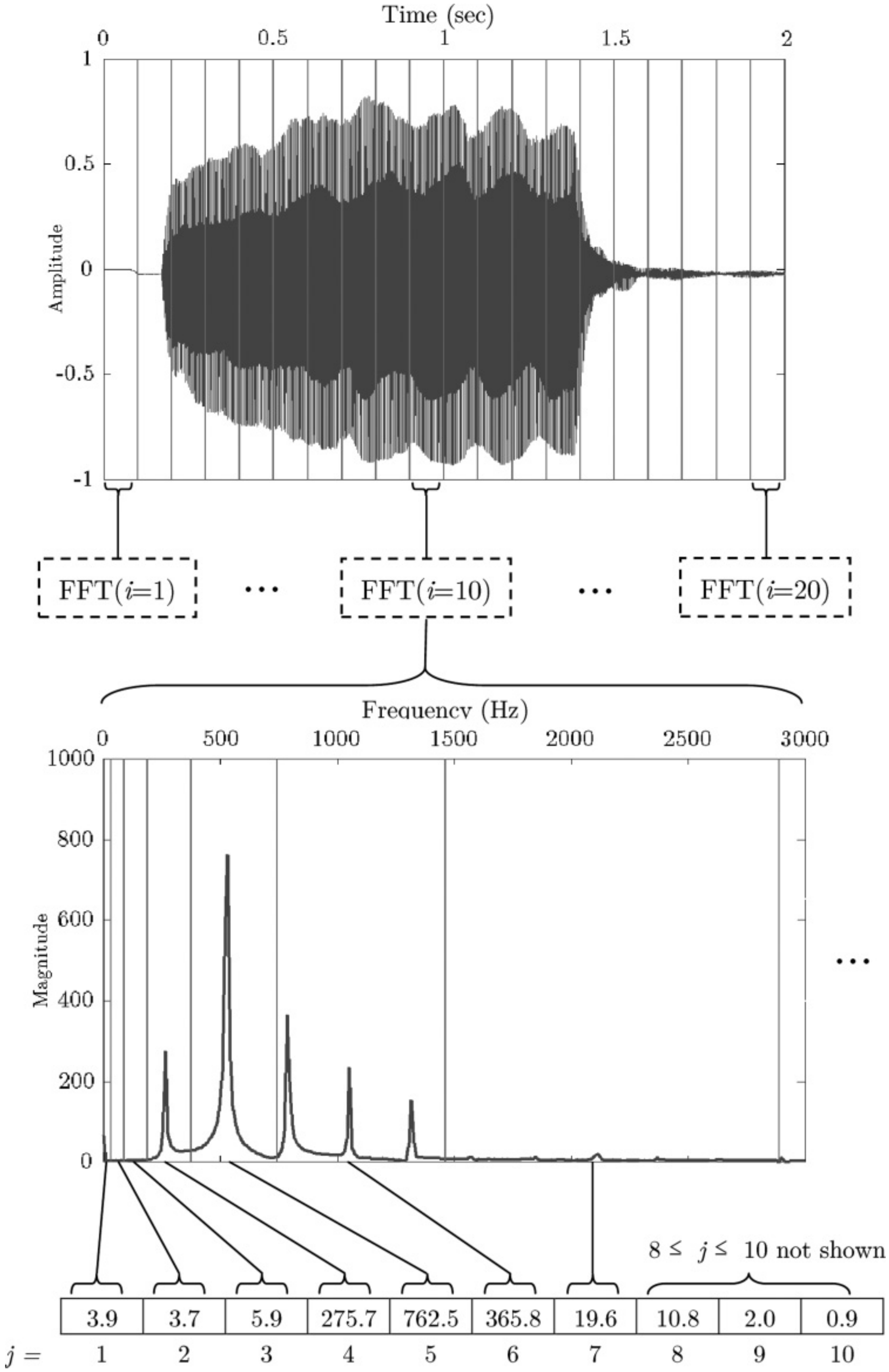
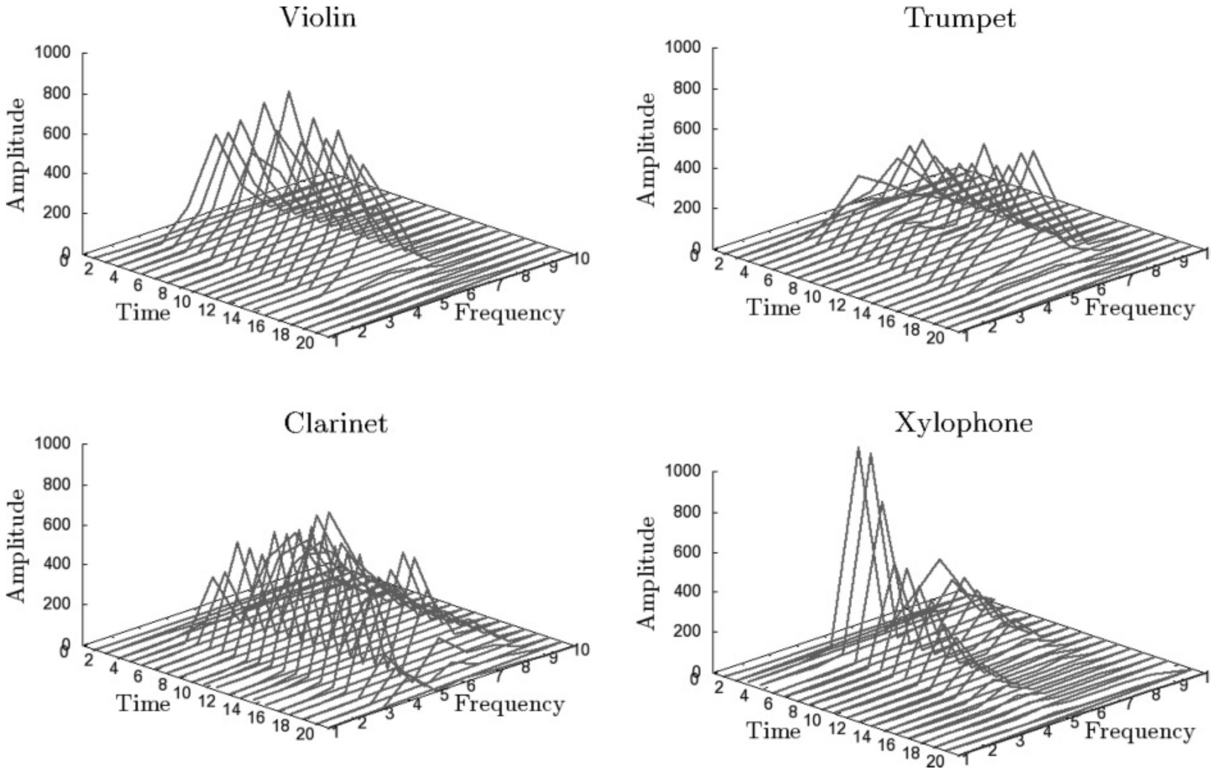


Figure 3. Visualization of the feature set for four different musical instruments each playing middle C at a mezzo forte dynamic level.


Violin

Trumpet

Clarinet

Xylophone

Naive Bayes

For a baseline Bayesian model, we chose the common naive 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 7 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 4a.

$$P(c \mid \mathbf{f}) = P(c) \times \prod_{f \in \mathbf{f}} P(f \mid c) \tag{7}$$

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 4b, denoted as $f_{j-1}^i \rightarrow f_j^i$. Equation 8a shows the class prior and the probability of the first row of the grid of features and Equation 8b defines the probability of the remaining features. There are no dependencies between the different time windows.

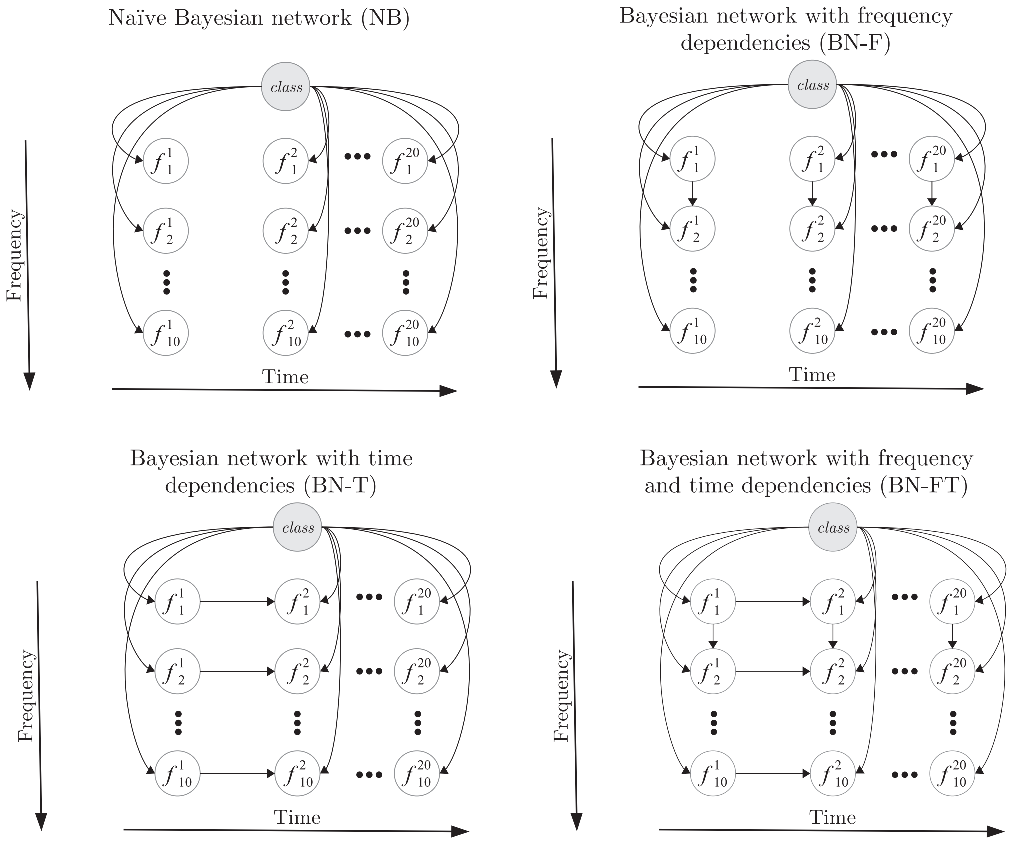
$$P(c \mid \mathbf{f}) = P(c) \times \prod_{i=1}^{20} P(f_1^i \mid c) \tag{8a}$$

$$\times \left(\prod_{i=1}^{20} \prod_{j=2}^{10} P(f_j^i \mid f_{j-1}^i, c) \right) \tag{8b}$$

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

Figure 4. Structure of the different Bayesian networks: a naive Bayesian network (NB) (a); a Bayesian network with frequency dependencies (BN-F) (b); a Bayesian network with time dependencies (BN-T) (c); and a Bayesian network with both frequency and time dependencies (BN-FT) (d). The arrows represent conditional dependencies between windows. In addition to the dependencies shown in the NB, the vertical arrows, such as those between f_1^1 to f_2^1 in BN-F, represent conditional dependencies between frequency windows. In BN-T, the horizontal arrows, such as those between f_1^1 to f_1^2, represent conditional dependencies between times windows. Finally, in BN-FT, the vertical and horizontal arrows represent conditional dependencies between both frequency and time windows, respectively.


Naïve Bayesian network (NB)
Bayesian network with frequency dependencies (BN-F)
Bayesian network with time dependencies (BN-T)
Bayesian network with frequency and time dependencies (BN-FT)
class
Frequency
Time
f_1^1
f_1^2
f_1^{20}
f_2^1
f_2^2
f_2^{20}
f_{10}^1
f_{10}^2
f_{10}^{20}

domain (see Figure 4c). Equation 9a shows the class prior and the probability of the first column of the grid of features and Equation 9b defines the probability of the remaining features.

$$P(c \mid \mathbf{f}) = P(c) \times \prod_{j=1}^{10} P(f_j^1 \mid c) \tag{9a}$$

$$\times \left(\prod_{i=2}^{20} \prod_{j=1}^{10} P(f_j^i \mid f_j^{i-1}, c) \right) \tag{9b}$$

Frequency and Time Dependencies

The final model, a Bayesian network with both time and frequency dependencies (BN-FT), is shown in Figure 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 10a shows the class prior and the probability of the upper-leftmost node (f_1^1) of the feature grid. Equation 10b shows the probability of first column of the grid, Equation 10c shows that of the first row of the grid, and Equation 10d shows the probabilities of the remaining features.

$$P(c \mid \mathbf{f}) = P(c) \times P(f_1^1 \mid c) \tag{10a}$$

$$\times \left(\prod_{i=2}^{20} P(f_1^i \mid f_1^{i-1}, c) \right) \tag{10b}$$

$$\times \left(\prod_{j=2}^{10} P(f_j^1 \mid f_{j-1}^1, c) \right) \tag{10c}$$

$$\times \left(\prod_{i=2}^{20} \prod_{j=2}^{10} P(f_j^i \mid f_j^{i-1}, f_{j-1}^i, c) \right) \tag{10d}$$

Baseline Algorithms

To explore the advantages of time and frequency dependencies between features, the accuracies of the grid-augmented Bayesian models are compared with two support vector machines, a k-nearest neighbor classifier, and naive Bayes. SVM and k-NN are 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 3) where $\delta = 2$ (SVM-Q). We also examined a radial basis function kernel and sigmoidal kernel. Both scored no better than an algorithm that randomly selects a solution, so these two kernels were not included in the subsequent 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.

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 yields a zero probability of the entire feature vector. To prevent this, we used the common technique of additive smoothing:

$$P(f_j^i) = \frac{x_i + \alpha}{N + \alpha \cdot d} \tag{11}$$

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 (Chen and Goodman 1996). 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.

Table 4. Classification Accuracy in Experiment 1

Algorithm	*Instrument*	*Family*
NB	81.57	80.94
BN-F	97.53	92.87
BN-T	96.36	94.39
BN-FT	**98.25**	97.09
SVM-L	81.46	85.57
SVM-Q	93.55	95.65
k-NN	92.99	**97.31**

Accuracy of classification (in percent), by instrument ($n = 24$) and by instrument family ($n = 4$), for the EastWest data set. Values in **boldface** indicate best results.

Experiments and Results

To test the utility of conditional dependencies between variables in the frequency and time realms, we conducted four experiments. In the first, we compared our Bayesian models against the baseline models on the EastWest data set in both the tasks of instrument identification and identification of musical instrument family. In the second experiment, we explored classification accuracy on instruments within the same musical family. In the third experiment we examine classification accuracy as a function of the number of instrument samples for each instrument. Lastly, in the the fourth experiment we examine the classification accuracy of all algorithms on the widely used Iowa data set.

Experiment 1: Instrument and Family Identification

The first experiment examined classification accuracy for both instrument identification ($n = 24$) and family identification ($n = 4$) on the EastWest data set. The results are shown in Table 4. The statistical significances using a paired student's t-test with $p \leq 0.01$ are shown in Table 5.

All of the Bayesian networks, with the exception of naive Bayes, outperformed both SVMs and k-NN

Table 5. Statistical Significance from Experiment 1

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/−	—

Statistical significance was measured using a 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.

in the instrument identification task. 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.

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 with 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 6. The Bayesian models showed increased confusion between brass and woodwind instruments compared to string or percussion instruments. The SVMs, *k*-NN, and naive Bayes, on the other hand, more often confused strings with either brass or woodwind compared with the Bayesian networks.

Experiment 2: Instrument Identification within Family

This experiment examines instrument classification by instrument family on the EastWest data set. Unlike Experiment 1, this experiment trains and tests only on instruments within the same family (see Table 7). The data set was divided into four separate data sets, 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 data sets.

Interestingly, the classification accuracy of strings, brass, and percussion exceeds 99 percent for all the Bayesian networks except naive Bayes, whereas woodwinds, the largest set of instruments ($n = 10$), achieves 97.86 percent accuracy. For the strings, brass, and percussion, BN-F and BN-FT achieve comparable accuracy. BN-FT outperforms BN-F on the more difficult woodwind set, however. The percussion set achieves the highest accuracy for all algorithms, includes the SVMs and *k*-NN. This result was expected, given the prominent attack of percussive instruments, and consistent with the literature.

Experiment 3: Accuracy by Data Set Size

This experiment examines the classification accuracy by instrument ($n = 24$) on the EastWest data set, similar to Experiment 1, but as the data set size varied from 100 to 1,000 in increments of 100 for each instrument (see Figure 5). The Bayesian network models converge to their respective optimal accuracy between 500 and 800 data samples per instrument. Nevertheless, 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

Table 6. Confusion Matrices for Experiment 1

		Classified as			
Algorithm	*Family*	String	Brass	Woodwind	Percussion
NB	String	**4,470**	21	327	162
	Brass	24	**3,021**	944	11
	Woodwind	277	1,923	**7,799**	1
	Percussion	220	320	324	**4,134**
BN-F	String	**4,865**	15	107	13
	Brass	3	**3,756**	239	2
	Woodwind	97	883	**9,009**	111
	Percussion	123	86	133	**4,658**
BN-T	String	**4,921**	0	34	45
	Brass	13	**3,612**	364	11
	Woodwind	173	600	**9,223**	4
	Percussion	27	55	21	**4,897**
BN-FT	String	**4,923**	3	67	7
	Brass	1	**3,627**	372	0
	Woodwind	19	198	**9,783**	0
	Percussion	4	15	13	**4,968**
SVM-L	String	**4,692**	11	254	43
	Brass	47	**1,265**	2,685	3
	Woodwind	140	226	**9,626**	8
	Percussion	25	3	19	**4,953**
SVM-Q	String	**4,670**	69	188	73
	Brass	84	**3,667**	245	4
	Woodwind	119	190	**9,680**	11
	Percussion	42	5	14	**4,939**
k-NN	String	**4,792**	56	107	45
	Brass	40	**3,795**	162	3
	Woodwind	43	145	**9,802**	10
	Percussion	22	6	6	**4,966**

The confusion matrices for family identification, using the EastWest data set, show classification counts. The row labels in the second column indicate the true instrument family. The column headers indicate the instrument family identified by the algorithm. Values in **boldface** indicate correct classifications.

examples per instrument. All the Bayesian models with dependencies achieved much higher accuracy with far fewer examples, however, than did either SVMs or *k*-NN. This important result will be useful when extending this system to real-world examples extracted from commercial audio recordings.

Experiment 4: Iowa Data Set

The final experiment examined classification accuracy for both instrument identification ($n = 25$) and family identification ($n = 3$) on the Iowa data set. The results are shown in Table 8. The statistical significances using a paired student's t-test with $p \leq 0.01$ are shown in Table 9.

As in Experiment 1, all of the Bayesian networks, again with the exception of naive Bayes, outperformed both SVMs and *k*-NN. The model with frequency dependencies outperformed the model with time dependencies. BN-FT and BN-F achieved comparable accuracies in the instrument task. The combination of both frequency and time dependencies outperformed BN-F and BN-T in the

Figure 5. Experiment 3: Accuracy of classification (in percent) on the EastWest data set, by number of examples per instrument for each model.

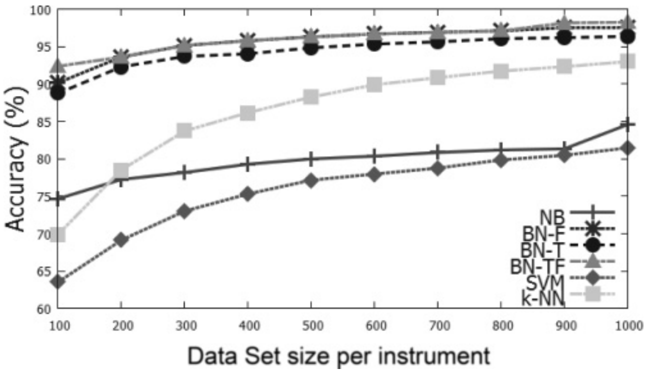


Table 7. Classification Accuracy in Experiment 2

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

Accuracy of classification (in percent), by instrument family ($n = 4$), for the EastWest data set. Values in **boldface** indicate best results.

Table 8. Classification Accuracy in Experiment 4

Algorithm	*Instrument*	*Family*
NB	46.34	73.30
BN-F	**80.76**	81.82
BN-T	75.25	81.24
BN-FT	**80.31**	87.33
SVM-L	65.36	75.03
SVM-Q	65.89	83.19
k-NN	72.78	**89.67**

Accuracy of classification (in percent), by instrument ($n = 25$) and by instrument family ($n = 3$), for the Iowa data set. Values in **boldface** indicate best results.

family identification task. Confusion matrices for the family identification task are shown in Table 10.

The Iowa data set contains fewer examples of each instrument compared to the EastWest data set, and several instruments in the Iowa data set contain less than 100 examples each. Nevertheless, these results on the Iowa data set are consistent with the our results on the EastWest data set when considering a smaller data set (see Figure 5).

Discussion

Many previous approaches, such as Agostini, Longari, and Pollastri (2003), 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 models, with the exception of naive Bayes, still outperformed both SVMs and *k*-NN on the woodwind data set in Experiment 2. 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 ($n = 4$) on the EastWest data set. The BN-F and BN-T models each achieved better accuracy on individual instrument classification than they achieved on family identification, however. 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. For both data sets, *k*-NN achieved higher accuracy on the family identification problem compared with the instrument identification problem. This is not surprising, because *k*-NN is known not to scale well as the number of classes increases (Jain and Kapoor 2009). Although the results of *k*-NN and the BN-FT model are competitive on the EastWest data set ($n = 4$), *k*-NN outperformed the BN-FT model on the family identification task on the Iowa data set ($n = 3$).

In examining the types of mistakes the algorithms made, we observed that the Bayesian models more often confused brass for woodwind instruments, compared with either strings or percussion (see Tables 6 and 10). This is perhaps unsurprising as our feature-extraction scheme sought to capture the conditional relationships of changes in amplitude

Table 9. Statistical Significance from Experiment 4

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

Statistical significance was measured 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 ($n = 25$) experiment and the second shows the family ($n = 3$) experiment.

Table 10. Confusion Matrices for Experiment 4

		Classified as		
Algorithm	*Family*	*String*	*Brass*	*Woodwind*
NB	String	**1,652**	425	450
	Brass	27	**403**	130
	Woodwind	99	76	**1,259**
BN-F	String	**2,013**	239	275
	Brass	12	**438**	110
	Woodwind	129	57	**1248**
BN-T	String	**1,962**	157	408
	Brass	17	**413**	130
	Woodwind	110	26	**1,298**
BN-FT	String	**2,256**	41	230
	Brass	35	**413**	112
	Woodwind	144	11	**1,279**
SVM-L	String	**2,293**	78	156
	Brass	225	**183**	152
	Woodwind	486	32	**916**
SVM-Q	String	**2,427**	41	59
	Brass	211	**286**	63
	Woodwind	338	48	**1,048**
k-NN	String	**2,303**	74	150
	Brass	18	**501**	41
	Woodwind	102	82	**1,250**

The confusion matrices for family identification, using the Iowa data set, show classification counts. The row labels in the second column indicate the true instrument family. The column headers indicate the instrument family identified by the algorithm. Values in **boldface** indicate a correct classification.

of frequencies over time. Woodwind and brass instruments are both classified as aerophones, instruments that generate sound by vibrating air, under the scientific classification of musical instruments devised by von Hornbostel and Sachs (1914).

As Deng, Simmermacher, and Cranefield (2008) 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 or MPEG-7 descriptors. Agostini, Longari, and Pollastri (2003) used a sparse set of nine spectral features to achieve 78.5 percent and 69.7 percent accuracy classifying 20 and 27 instruments, respectively, using an SVM. Our feature-extraction scheme, using 200 time- and frequency-varying features, achieved 93.6 percent accuracy classifying 24 instruments also using an SVM.

Although not directly comparable, these results imply that our feature-extraction scheme better handles a larger number of instrument classes. Although our system uses 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.

The goal of this article was to explore the utility of statistical dependencies of the features in both the time and frequency domains. In these experiments, the structure of the Bayesian models are tied to the feature-extraction scheme employed. Therefore it is not possible to compare our feature-extraction scheme to other schemes common in the literature using the Bayesian networks. Our experiments independently demonstrated the success of Bayesian classifiers on both the EastWest and Iowa data sets. Livshin and Rodet (2003) noted the importance of cross-database comparison. The examples in the Iowa data set are longer in duration than those in the EastWest data set. Because our feature-extraction scheme relies on temporal and frequency partitions, a cross-database comparison is not possible as the features do not align between the two data sets. This was expected and validated by preliminary experiments. Our classification technique is successful on both data sets independently, but unsuccessful on a cross-data set experiment. These results imply the success of our method relies on the classification technique.

Anecdotally, our results, when compared with previously published results, indicate the value of our feature-extraction scheme's ability to define statistical dependencies between features. Perhaps 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.

Future Work

Given the success of the Bayesian networks for the monophonic, single instrument classification problem, future experiments will attempt to reduce the number of features used to train the model. The number of time windows will be examined empirically to determine the minimum number of time windows required to maintain comparable classification accuracy. First, any time windows that capture the decaying resonance of the instrument sample could potentially be discarded. A visual inspection of Figure 3 indicates there is opportunity to prune the feature space, such as eliminating the unnecessary frequency windows that contain data below 20 Hertz, the lower threshold of human hearing. Secondly, motivated by the observation that the frequency model outperformed the time model in all experiments, the size of each time window can be increased and the minimum number of time windows necessary will be empirically tuned. In future experiments, we aim to refine our feature-extraction scheme to allow cross-data set comparisons, such as testing extraction schemes that do not rely on temporal partitioning, allowing comparison of feature-extraction schemes.

This Bayesian approach to musical instrument classification might be expanded to multi-label classification, such as multiple instrument classification in cases where multiple monophonic instruments are playing simultaneously. Producing training examples of every possible combination of musical instruments is infeasible, even just for pairs of instruments, let alone combinations of three or more. Therefore, we will explore transforming the problem into a set of the binary relevance classifiers (Tsoumakas and Katakis 2007) in which the current models will be trained using single instruments and then used in series to attempt to classify test data with multiple instruments sounding at the same time. To achieve multi-label classification, separate binary models will be trained for each instrument to classify whether or not that instrument is present in the example. Then, for each test example, the set of instruments that are classified as present in the signal will be returned.

Conclusion

In this work, we have presented an approach to 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 naive 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.

References

Agostini, G., M. Longari, and E. Pollastri. 2003. "Musical Instrument Timbre Classification with Spectral Features." *EURASIP Journal on Applied Signal Processing* 2003:5–14.

Bagwall, C., et al. 2013. "Sound eXchange." Available online at sox.sourceforge.net. Accessed May 2013.

Barbedo, J. G. A., and G. Tzanetakis. 2011. "Musical Instrument Classification Using Individual Partials." *IEEE Transactions on Audio, Speech, and Language Processing* 19(1):111–122.

Benetos, E., M. Kotti, and C. Kotropoulos. 2006. "Musical Instrument Classification Using Non-Negative Matrix Factorization Algorithms." In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1844–1847.

Boser, B., I. Guyon, and V. Vapnik. 1992. "A Training Algorithm for Optimal Margin Classifiers." In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152.

Burred, J. J., A. Robel, and T. Sikora. 2009. "Polyphonic Musical Instrument Recognition Based on a Dynamic Model of the Spectral Envelope." In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 173–176.

Burred, J. J., A. Robel, and T. Sikora. 2010. "Dynamic Spectral Envelope Modeling for Timbre Analysis of Musical Instrument Sounds." *IEEE Transactions on Audio, Speech, and Language Processing* 18(3):663–674.

Chen, S., and J. Goodman. 1996. "An Empirical Study of Smoothing Techniques for Language Modeling." In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pp. 310–318.

Cover, T., and P. Hart. 1967. "Nearest Neighbor Pattern Classification." *IEEE Transactions on Information Theory* 13(1):21–27.

Deng, J., C. Simmermacher, and S. Cranefield. 2008. "A Study on Feature Analysis for Musical Instrument Classification." *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 38(2):429–438.

Donnelly, P., and C. Limb. 2009. "Music." In L. R. Squire, ed. *Encyclopedia of Neuroscience*. Amsterdam: Elsevier, pp. 1151–1158.

EastWest. 2013. "Quantum Leap Symphonic Orchestra Library." Available online at www.soundsonline.com/Symphonic-Orchestra. Accessed March 2012.

Eronen, A. 2003. "Musical Instrument Recognition Using ICA-based Transform of Features and Discriminatively Trained HMMs." In *Proceedings of the Seventh International Symposium on Signal Processing and Its Applications*, vol. 2, pp. 133–136.

Essid, S., G. Richard, and B. David. 2006. "Musical Instrument Recognition by Pairwise Classification Strategies." *IEEE Transactions on Audio, Speech, and Language Processing* 14(4):1401–1412.

Fayyad, U., and K. B. Irani. 1992. "On the Handling of Continuous-valued Attributes in Decision Tree Generation." *Machine Learning* 8(1):87–102.

Fayyad, U., and K. B. Irani. 1993. "Multi-Interval Discretization of Continuous-valued Attributes for Classification Learning." In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1022–1027.

Friedman, N., D. Geiger, and M. Goldszmidt. 1997. "Bayesian Network Classifiers." *Machine Learning* 29(2):131–163.

Fujinaga, I., and K. MacMillan. 2000. "Realtime Recognition of Orchestral Instruments." In *Proceedings of the International Computer Music Conference*, pp. 241–243.

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

Heittola, T., A. Klapuri, and T. Virtanen. 2009. "Musical Instrument Recognition in Polyphonic Audio Using Source-Filter Model for Sound Separation." In *Proceedings of the International Society for Music Information Retrieval*, pp. 327–332.

Jain, P., and A. Kapoor. 2009. "Active Learning for Large Multi-Class Problems." In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 762–769.

Jelinek, F. 1997. *Statistical Methods for Speech Recognition*. Cambridge, Massachusetts: MIT Press.

Joder, C., S. Essid, and G. Richard. 2009. "Temporal Integration for Audio Classification with Application to Musical Instrument Classification." *IEEE Transactions on Audio, Speech, and Language Processing* 17(1):174–186.

jVSTwRapper. 2013. "Java-based Audio Plug-Ins." Available online at jvstwrapper.sourceforge.net. Accessed March 2012.

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

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

Liu, J., and L. Xie. 2010. "SVM-Based Automatic Classification of Musical Instruments." In *Proceedings of the International Conference on Intelligent Computation Technology and Automation*, vol. 3, pp. 669–673.

Livshin, A., and X. Rodet. 2003. "The Importance of Cross Database Evaluation in Sound Classification." In *Proceedings of the International Symposium on Music Information Retrieval*, pp. 241–242.

Marques, J., and P. Moreno. 1999. "A Study of Musical Instrument Classification Using Gaussian Mixture Models and Support Vector Machines." CRL Technical Report Series 4. Cambridge, UK: Cambridge Research Laboratory.

Native Instruments. 2013. "Kontakt." Available online at www.native-instruments.com/en/products/komplete/synths-samplers/kontakt-5. Accessed March 2012.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning*, vol. 1. San Mateo, California: Morgan Kaufmann.

Tsoumakas, G., and I. Katakis. 2007. "Multi-Label Classification: An Overview." *International Journal of Data Warehousing and Mining* 3(3):1–13.

University of Iowa. 2013. "Musical Instrument Samples." Available online at theremin.music.uiowa.edu/MIS.html. Accessed May 2013.

Vaill, C. 2013. "Normalize." Available online at normalize.nongnu.org. Accessed May 2013.

Vapnik, V. 1999. *The Nature of Statistical Learning Theory*. Berlin: Springer.

von Hornbostel, E. M., and C. Sachs. 1914. *Systematik der Musikinstrumente*. Berlin: Behrend. Translated by A. Baines and K. P. Wachsmann as "Classification of Musical Instruments." 1961. *Galpin Society Journal* 14:3–29.

Wieczorkowska, A. 1999. "Classification of Musical Instrument Sounds Using Decision Trees." In *Proceedings of the International Symposium on Sound Engineering and Mastering*, vol. 99, pp. 225–230.