2008-01-01

Violin Sound Quality Detection

Jane Charles
Technological University Dublin

Derry Fitzgerald
Cork Institute of Technology

Eugene Coyle
Technological University Dublin

Recommended Citation

Charles, J.A. and Fitzgerald, D. & Coyle, E. Violin sound quality detection. Paper given at the at the *Irish Signals and Systems Conference, Galway, 2008.*

Violin Sound Quality Detection

J. A. Charles*, D. Fitzgerald, E. Coyle***

**School of Electrical Engineering Systems*
Dublin Institute of Technology Dublin 8
email : violincharles@gmail.com

***Department of Electronic Engineering*
Cork Institute of Technology
email: derry.fitzgerald@cit.ie

***Abstract*— Although much research has been completed on finding features for instrument recognition systems, little work has focused on the violin's timbre space. Suitable features from which a computer can assess the quality of a violinist's playing have been sought and the classification of violin note sound quality is investigated in this paper. The eventual outcome of this work can be applied in various systems including the development of a violin or bowed string instrument teaching aid, in automatic music transcription and information retrieval or classification systems.**

***Keywords* – violin, timbre, k-means clustering, nearest-neighbour (NN) classification**

I INTRODUCTION

The development of a computer based violin teaching aid was proposed in [1]. In order to begin to define a violin timbre space, the relationship between playing technique and sound produced must be understood and quantified. This is to allow for guideline 'boundaries' to be associated with not only good violin sound but also with poorer or beginner violin playing. The more general area of quantifying the qualitative and subjective nature of violin playing using signal processing techniques was presented in [2, 3]. This has enabled the representation of violin sounds by suitable descriptors. Fifteen features are used to define the violin sounds in this work. Violin playing faults have been identified and are limited to nine faults [2]. This paper considers the classification of violin notes. Two tasks are put to the classifier: the first is the detection of beginner note from professional standard note and the second is much more specific involving individual fault detection. The aim of this work is to test the feasibility of developing a home computer based violin teaching tool.

II EXISTING RESEARCH

Within the broad area of automatic audio classification, much work has been done in speech recognition, in discriminating between speech and non-speech sounds, and in instrument and environmental sound identification and synthesis. Much of the existing research to do with violins has been carried out to in order to better understand and emulate the making of top quality sounding instruments. Many methods have been applied to gain insight into the complex interactions between the various components of stringed instruments. Work exploring the effect a violin player has on the sound produced is limited. Many features, although very useful in determining one instrument from another [4, 5], are not appropriate for catching the subtleties due to playing technique or for use within the violin's timbre space. Current advances in signal processing and interactive computing have enabled the development of much more sophisticated systems and learning aids, such as that which has been demonstrated by Hämäläinen *et al.* [6]. This successful real-time singing aid involves pitch-based control of a game character by the user's voice. Little work has been conducted on characterizing or describing the violin's timbre space let alone exploring the relationship between timbre and playing technique.

III DATA TEST SET

As no suitable data set was readily available, one had to be made. Much thought was given in creating this data set in terms of what was needed, obtainable and viable. The ideal data set would be a type of violin timbre real sound continuum. Unfortunately, this would be very time consuming, if not near impossible to obtain. The first bow stroke a beginner must learn is called *legato*, which literally means 'tied together' or smoothly connected [7]. Mastering this ensures enough bow control upon which the student can develop other bow strokes, such as *staccato* ('disconnected'). Since the style or type of bow stroke used effects the readings obtained, only professional standard player *legato* notes will be used and the beginner notes will be compared to these.

The data test set was created in a controlled environment, and consists of two same sized groups, one with beginner notes and the other with professional

standard ‘good’ player legato notes. The samples all contain one note and are of varying lengths and pitches. The pitch range of the data set is any note played in the ‘first position’, which is the lowest possible position on the violin, i.e. open G3 to B5, fourth finger on the E string. Two professional standard players and three beginner players made recordings from which the data set was collected. A player will never play two notes exactly the same although they may be perceived by a listener as being the same. A beginner does not have the control necessary to achieve this level of accuracy in playing. Hence, it is more appropriate to use features which do not dependent on ether note length or pitch. The data samples were made in a recording studio using four microphones, a directional stereo pair, and two omni directional microphones. The tracks were recorded onto DAT, mixed and saved as monophonic wav files. It should also be noted that the recordings were all made using the same set up and the same violin and bow. There are eighty-eight beginner note samples and eighty-eight legato ‘good’ note samples.

IV Data Set Feature Vectors

Each sample in the data set is represented by its feature vector. The features selected to characterise the samples are based on their ability to separate the data into distinct groups within their respective domains [2, 3]. The fifteen features chosen are numbered as shown in Table 1.

Number	Feature
1	time domain mean
2	Mel cepstrum variance
3	real cepstrum coefficients mean
4	real cepstrum coefficients variance
5	real cepstrum coefficients kurtosis
6	1st real cepstrum coefficient
7	2nd real cepstrum coefficient
8	5th real cepstrum coefficient
9	spectral contrast measure
10	spectral flatness measure
11	spectral flatness variance
12	spectral flatness std deviation
13	spectral flatness skew
14	signal average power
15	autocorrelation coefficient

Table 1: Features Used.

The data set is represented by a 176 x 15 array, where 176 is the number of samples.

V Listening Tests

Listening tests have been included to remove the subjective nature of this research by showing that other trained string players can hear and recognize the faults and sound quality descriptions. From the results of these listening tests, it is hoped that a relationship can be established between what people perceive and any quantitative features for the sound samples. These tests are aimed at professional standard violinists in particular but, to increase numbers, cellists and violists have also been included. The listening group consisted of twenty-one string players. The listeners received no training, only a copy of the testing process steps and an explanation of the terms. A play list which includes all the beginner and legato good note samples, 176 samples in total, exists. As soon as the listener activates the testing/listening program, a random play list is generated consisting of all samples from the list. After having heard the note, the listener selects the terms which best characterise the sound and grades the overall quality. The sound characteristics list includes descriptions of playing faults and the overall sound quality is a grade between 1 (very poor) and 6 (excellent). The faults or sound characteristics are described and numbered in Table 2.

Number	Fault Name
Fault 1	crunching
Fault 2	skating
Fault 3	nervousness
Fault 4	intonation
Fault 5	bow bouncing
Fault 6	extra note
Fault 7	sudden end to note
Fault 8	poor start to note
Fault 9	poor finish to note

Table 2: Fault Descriptions.

The listener was also left space to add their own comments. The exact play list for each listener only becomes available at the end of the listening test. The test progresses at a speed controlled by the user and each sample can only be played once. AKG K240 ‘Monitor’ (600 Ohms) headphones were used and samples were accessed and played through Matlab. The consistency of the results obtained from this test were checked and found to be acceptable. Normalising these results allowed for an ‘average listener’ to be established. This ‘average listener’ is what is used for investigating how violin timbre is perceived and for a priori sample labelling.

VI A Priori Sample Labelling

Two groups of labels have been obtained: one considering the overall sound quality and the other for individual faults perceived. These labels reflect the normalised listeners’ perception, which has been obtained from the listening tests. The listeners had to evaluate the overall sound quality of all samples by giving a grade between 1 (very poor) and 6 (excellent) and by indicating playing faults perceived. As only two clusters are required, class labels of 1 for professional player notes and 2 for beginner notes need to be assigned, reflecting the listeners’ perception. This was done by finding all the samples which had been given a grade of 5 or above and re-labelling them as 1s and the remaining samples as 2s. Grading level 5 was taken and not 4 because only the good to excellent sounds should be classified as professional sounds and not those with quality perceived as being ‘reasonable’. The data set

consists of eighty-eight beginner notes and eighty-eight legato professional standard notes and certain notes which had been played by the beginner were perceived as good sounds by the listeners. 82 of the 176 samples were perceived as good and consequently have been labelled '1' and the remaining 94 have label '2'. Using the information obtained about fault perception, labels were assigned according as to whether a fault had been perceived or not. Samples perceived to have a specific fault have been labelled with 2s and for the fault not having been perceived, 1s. However, faults rarely occur in isolation and many of the beginner player samples contain more than one fault.

VII CLASSIFICATION

Classification is the general term given to organizing or grouping similar data together according to selected characteristics or some common feature. Grouping data together based on similar patterns or descriptive features allows a class label to be associated with the group. The most significant aims of classification relate to data simplification and prediction, increasing the efficiency of tasks such as information retrieval [8]. In this paper, the classification of note samples into beginner or professional and fault identification are tested. The aim is to provide objective and stable classification for the subjective nature of violin sounds for possible ultimate use in a computer based teaching aid.

The first stage of the classification process involves clustering which is used to find centres that reflect the distribution of data points [9]. Running the k-means clustering algorithm provides the prototype vectors which are then used in the k-NN classifier. Although many clustering methods exist, k-means is one of the most often used because of its simplicity and converges well with the Euclidean distance which is given in equation 1 [9].

$$dist = \frac{1}{N}\left(\sum_{n=1}^{N} (vectA(n) - vectB(n))^2 \right)^{1/2} \qquad (1)$$

One advantage of using the Euclidean distance is that each feature remains equally important and no correlations between variables influence the outcome. The k-means clustering code, taken from the Somtoolbox [10], uses the iterative partitional clustering algorithm put forward by Jain and Dubes, a description of which can be found in [9]. An advantage of this algorithm is that it automatically assigns items to clusters. The disadvantages are that the number of clusters must be pre-selected and that all items are forced into a cluster, making it very sensitive to outliers. The squared Euclidean distance metric is used which is computationally faster for clustering than the Euclidean distance shown in equation 1. The clustering algorithm remains unaffected by this change, as it is a partitional clustering method, as opposed to a hierarchical one.

For the first task, two clusters are sought: one for poorer quality sounds and another for professional violinist notes. The 'beginner' and the 'professional' clusters provide the k-NN classifier with its prototype vectors. They are two 15 x 1 vectors. For the fault identification task, clusters are formed according to the presence or absence of a particular fault as perceived by the listener. Prior to use in the classifier, these cluster vectors were checked by comparing their values with the means of all samples for each feature associated with its respective cluster. The algorithm converged well and no alterations had to be made.

The data set's features are stored in a 176 by 15 array, where 176 is the total number of samples and 15, the number of features. A proximity matrix is then calculated using the squared Euclidean measure between the prototypes and each feature vector. This matrix is inputted into the k-NN classifier, to which class labels are assigned. These labels are then compared with the a priori labels to obtain the classifier accuracy reading. Classifier accuracy is the probability of correctly labelling a randomly selected sample. The k-NN rule classifies a sample by assigning it the label which is most often associated with its k-nearest samples. When k=1, every sample is assigned to the class of the nearest cluster or pattern. In practice, k=1 is often used, as it is in this work.

Should the classification process be carried out on the entire data set, very specific model building information will be obtained. Cross-validation techniques are methods for detecting and preventing classifier over-fitting, checking classifier accuracy estimation and generalisation potential. It is a way of ensuring that a classifier can perform in an unsupervised situation. To conduct cross-validation, the data set is put in a random order after which, a portion of the data set is put aside as a 'training' set and leaving the rest for testing. Two well established cross-validation techniques are n-folds and leave-one-out cross-validation (LOOCV). In n-fold cross-validation, the data set in put into n equal sections where n-1 sections are used for training and the remaining section for testing. In LOOCV, as the name implies, each sample is removed one at a time and used for testing and the rest of the samples are used for training. This makes LOOCV an almost unbiased method but high variance can be a problem which can lead to unreliable estimates [11]. From a purely practical perspective, LOOCV is computationally intensive and is better used on smaller data sets and also the deciding factor in using four-fold cross-validation in this work.

IX RESULTS

Classification results obtained are based on four-fold cross validation to minimise classifier over-fitting.

In four-fold cross validation, the randomly ordered samples are divided into four equal parts. Radomising the data set prior to dividing it up reduces the possibility of biasing the cross-validation. Three quarters of the data set is used for training and the remaining quarter for testing the classifier. This is done in rotation so that each quarter is used as the test set once. The results are compared and the error readings are checked. The error readings indicate the difference in classifier performance between the training and testing sets. The smaller the error, the better the associated conditions or feature choice suits the classification task. This procedure can be repeated using different initial random data set orderings for further verification. Four-fold cross validation has been applied to both tasks and all possible feature combinations. The training and testing set means across all four folds are used and are shown below. Successful result summaries for the beginner versus professional task can be seen in Table 3 and for individual fault detection in Table 4.

No. features used	Train	Test	Features
3	95.45%	95.45%	1, 6, 9
3	95.45%	95.45%	6, 8, 9

Table 3: Training and Testing Set Means for Detection of Beginner Notes from Professional Notes.

Fault	No. features used	Train	Test	Features
5	3	84.28%	86.08%	3, 8, 14
3	2	76.57%	77.84%	1, 2
3	4	76.52%	76.14%	3, 7, 8, 10

Table 4: Successful Fault Detection Training and Testing Set Means.

These tables provide only a brief summary of the most successful and efficient results. The results will be dealt with in greater detail in their respective sections next.

a) Overall Sound Quality Detection

The first task is to detect good sound from poorer sound quality such as that associated with a beginner. This involves looking at the accuracy achieved by both training and testing sets for every possible feature combination, a summary of which is shown in Table 5, where the leftmost column indicates the number of features used. The next two columns show the top accuracy readings achieved for training and testing sets. The fourth column gives the number of combinations achieving the relevant accuracy scores. The rightmost column compares the combinations obtained for the testing and training sets which have returned top accuracy readings.

No.	Top Train	Top Test	No. Combinations	Train = Test?
1	75.57%	75.57%	14(train), 15 (test)	inclusive
2	93.75%	93.75%	1	yes
3	95.45%	95.45%	2	yes
4	95.45%	95.45%	2	yes
5	95.45%	95.45%	11	yes
6	95.45%	95.45%	13	yes
7	95.45%	95.45%	33	yes
8	95.45%	95.45%	18	yes
9	95.45%	95.45%	19	yes
10	95.45%	95.45%	10	yes
11	95.45%	95.45%	2	yes
12	75.57%	75.57%	15(train), 17(test)	inclusive
13	75.57%	75.57%	4(train), 5(test)	inclusive
14	75.57%	75.57%	1	yes
15	75.57%	75.57%	1	yes

Table 5: Summary of Top Results for Training and Testing Sets According to Number of Features Used for Task I.

According to these results, automatic detection between good and beginner notes can be done effectively and easily using three up to eleven features. A small drop in accuracy readings is reported when less than two features are used and a drop of ≈20% is observed when twelve or more features are used.

The results providing the top accuracy reading using the least amount of features are of greatest interest and can be seen in Table 6.

no. features used	train	test	combination
2	93.75%	93.75%	1, 12
3	95.45%	95.45%	1, 6, 9
3	95.45%	95.45%	6, 8, 9
4	95.45%	95.45%	1, 2, 6, 9
4	95.45%	95.45%	2, 6, 8, 9

Table 6: Top Performing Feature Combinations Using Two, Three and Four Features Only.

In Table 6, only the top results obtained using two, three or four features are shown. The successful three and four feature combinations only differ by the addition of one feature, feature two. This makes feature two redundant when four features are used. Features numbered six and nine are present in both combinations. Using these two features only in the classifier returned a reading of ≈54% accuracy. So adding feature one or feature eight greatly improves the accuracy reading. There is no evident relationship or interdependence between features one and eight. Feature one has been shown to be 100% accurate in detecting these two groups on its own [3] but slightly less efficient at 75.57% through a classifier with monothetic clusters. The top performing combination when using two features obtained 93.75% accuracy, using features one and twelve.

Using five to ten feature combinations increases the number of successful combinations and feature overlap needs to be considered. A first general comment about all the top scoring combinations is that

they contain features six and nine. When considering only feature combinations obtaining accuracy results greater than 90%, there are one hundred and eleven successful combinations using from two to eleven features inclusively. Of these combinations when the redundant combinations have been removed, there are but eighteen combinations. With redundancy eliminated, it is possible to use two, three, five, six, seven, or eight feature dependent combinations to achieve greater than 90% accuracy. Apart from the four feature exception, it is only after adding a ninth feature that redundant features are present in every top scoring combination. It is more efficient to use the smallest number of features to achieve classification. The task of detecting beginner from professional standard notes is possible at 95.45% accuracy and using three features is the most efficient way of determining good from beginner notes.

b) Individual Fault Detection

The second task involves individual fault detection. Table 7 shows the best scores achieved for detecting each fault. Some feature combinations are successful at fault detection but the same combinations often detected more than one fault successfully, making individual fault detection difficult to achieve. Using four, five or six feature combinations return the best accuracy scores for all faults except for fault three. The same feature combinations are returned and can be seen in Table 8.

Fault	Train	Test	No. features used
1	78.88%	77.56%	4, 5, 6
2	80.59%	82.67%	4, 5, 6
3	76.52%	76.14%	2
4	82.20%	81.25%	4, 5, 6
5	87.12%	86.36%	4, 5, 6
6	87.12%	89.20%	4, 5, 6
7	81.91%	79.26%	4, 5, 6
8	82.58%	81.82%	4, 5, 6
9	75.76%	74.43%	4, 5, 6

Table 7: Top Detection Scores for Each Fault.

No. features used	Features
2	1, 2
4	1, 8, 10, 13
5	1, 3, 8, 10, 13
6	3, 4, 8, 11, 12, 13
6	4, 8, 10, 11, 13, 15

Table 8: Successful Feature Combinations from Table 7.

A summary of the top results achieved according to the number of features used can be seen in Table 9 where the leftmost column gives the number of features used.

No.	Train	Test	Fault	Features
1	≈ 50%	≈ 50%	no	n/a
2	76.52%	86.08%	3	1, 2
3	84.28%	86.36%	5	3, 8, 14
4	87.12%	89.20%	5	1, 8, 10, 13
4	87.12%	86.36%	6	1, 8, 10, 13
5	87.12%	89.20%	5	1, 3, 8, 10, 13
5	87.12%	86.36%	6	1, 3, 8, 10, 13
6	87.12%	89.20%	5	3, 4, 8, 11, 12, 13
6	87.12%	86.36%	5	4, 8, 10, 11, 13, 15
6	87.12%	89.20%	6	3, 4, 8, 11, 12, 13
6	87.12%	89.20%	6	4, 8, 10, 11, 13, 15
7	84.38%	86.08%	5	1, 4, 6, 7, 9, 14, 15
7	84.38%	86.08%	5	3, 4, 9, 11, 12, 14, 15
7	84.38%	86.08%	5	3, 7, 8, 11, 12, 13, 14
8	84.38%	86.08%	5	3, 6, 7, 8, 9, 11, 13, 14
9	84.38%	86.08%	5	1, 2, 3, 4, 6, 7, 8, 14, 15
9	84.38%	86.08%	5	1, 2, 3, 6, 8, 9, 10, 12, 14
9	84.38%	86.08%	5	1, 6, 7, 8, 9, 10, 12, 13, 14
9	84.38%	86.08%	5	3, 4, 6, 7, 8, 10, 11, 12, 14
9	84.38%	86.08%	5	3, 4, 6, 7, 9, 11, 12, 14, 15
10	84.38%	86.08%	5	4, 7, 8, 9, 10, 11, 12, 13, 14, 15
11	84.38%	86.08%	5	1, 2, 3, 6, 7, 8, 10, 11, 12, 13, 14
12	76.33%	76.99%	5	1, 2, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14
13	75.76%	78.13%	5	1, 2, 3, 4, 6, 7, 8, 9, 10, 12, 13, 14, 15
14	≈ 50%	≈ 50%	no	n/a
15	≈ 50%	≈ 50%	no	n/a

Table 9: Faults Detected According to the Number of Features Used.

From Table 9, fault detection is not possible using one, fourteen or fifteen feature combinations as the results obtained for all of these combinations are inconclusive at ≈50% accuracy. The results of greatest interest are those obtained when using two, three, seven, eight, nine, ten and eleven feature combinations. From inspection of all the top scores obtained, fault five is the easiest to detect. Whether using three through to thirteen features, fault five is consistently detected with the highest levels of accuracy. Using four, five or six feature combinations also detects fault six with the same high accuracy and features as fault five as can be seen in Table 9. Faults five and six are bow bouncing and extra note respectively.

Using three, seven, eight, nine, ten or eleven feature combinations all provide solutions for detecting fault five with mean accuracy above 80%. A summary of these results is given in Table 10.

No.	Train	Test	Total no.	No. dependent
3	84.28%	86.08%	1	1
7	84.38%	86.08%	3	2
8	84.38%	86.08%	1	0
9	84.28%	86.08%	5	2
10	84.38%	86.08%	1	1
11	84.28%	86.08%	1	1

Table 10: Fault Five Detection Results.

In Table 10, the leftmost column indicates the number of features used. Feature combinations which have performed poorly or overlap in the detection of another fault have been omitted. The training and testing set mean scores are shown in the next columns. The fourth

column gives the total number of useful combinations and the last column has the number of successful combinations without overlapping samples. Although fault five can be detected using seven, nine, ten or eleven feature combinations, the most efficient way to detect fault five uses three features, three, eight and fourteen.

Fault three is the next easiest fault to detect as it all but once returns completely different feature combinations than any other fault. Fault three is most readily detected when using two features. The training and testing sets achieved 76.52% and 76.14% accuracy respectively. These are the top accuracy readings obtained for any two feature combination. The other faults which returned their top scores with the same combination are all at least ≈10% lower. The next best performance using two features was for detecting fault five with readings of 69.41% and 66.19% for the training and testing sets respectively. These results though are achieved with a different combination, one which uses features four and fourteen. Looking at how successfully fault three is detected by other combinations revealed an interesting pattern. It was one fault which achieved its highest accuracy rates using completely different feature combinations to all other faults.

X CONCLUSIONS

Detecting good sound from beginner sound can be achieved by any one of a hundred and ten different feature combinations, using from three to eleven features, returning accuracy results of just below 96%. On further investigation, much feature redundancy is present. Removing overlapping feature combinations leaves only eighteen all feature dependent combinations. The most efficient way to detect a beginner from a professional violin sound is to use three features.

The presence of playing faults can be detected successfully as can be seen in the results shown Table 7. Individual faults though are harder to isolate. Only two specific faults are easy to detect. They are fault three, which is 'nervousness' and fault five which is 'bow bouncing'. The detection accuracy rates for the other faults are all closely grouped together, and return the same feature combinations, implying a certain qualitative proximity from a quantitative perspective. This is due in part to a sonic similarity between certain faults and that the fault samples in the data set often contain more than one fault. One possible way around this would be to use samples which contain only one fault at a time. Difficulties relating to data set creation have already been mentioned in Section III. An alternative to changing the data set would be to pay greater attention to the naming of faults, making them more specific. Another point to investigate would be to find new features. Location dependent features, such as those pertaining to the attack and end of note periods could be more informative.

The results for both tasks have been obtained via cross validation on one data set. It remains to be confirmed whether they hold on a different data set.

REFERENCES

[1] Charles, J. A. *et al.* 'Towards a Computer Assisted Violin Teaching Aid', International Symposium on Psychology and Music Education, Nov. 29-30, 2004, Padua, Italy.

[2] Charles, J. A., *et al.* 'Quantifying Violin Timbre', DMRN, London, 2006.

[3] Charles, J. A., *et al.* 'Violin Timbre Space Features', ISSC'06, Dublin Institute of Technology, Dublin, June 28-30, 2006.

[4] Eronen, A., Klapuri, A., 'Musical Instrument Recognition Using Cepstral Coefficients and Temporal Features', Signal Processing Lab, Tampere University of Technology, Tampere.

[5]Martin, K. D., Kim, Y. E., 'Musical Instrument Identification: A Pattern-Recognition Approach', 136th Meeting ASA, Oct. 1998.

[6] Hämäläinen, P., Mäki-Patola,T., Pulkki, V., Airas, M. 'Musical Computer Games Played by Singing', Proc. 7th Int. Conf. on Digital Audio Effects (DAFx'04), Naples, Oct. 5-8, 2004.

[7] Jackson, B. G., Berman, J., Sarch, K. *The ASTA Dictionary of Bowing Terms for Stringed Instruments*, American String Teachers Association, 3rd edition, Tichenor Publishing Group, Bloomington, Indiana, 1987.

[8] Bishop, C. M., *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1996.

[9] Jain, A. K., Dubes, R. C., *Algorithms for Clustering Data*, Prentice-Hall, 1988.

[10] www.cis.hut.fi/projects/somtoolbox/

[11] Makatou, M., Tian, H., Biswas, S., Hripcsak, G., 'Analysis of Variance of Cross-Validation Estimators of the Generalized Error', Journal of Machine Learning Research 6, 1127-1168, July 2005.