# Musical style recognition - a quantitative approach

Peter van Kranenburg
Faculty of Arts, University of Utrecht, Netherlands
p.vankranenburg@lodebar.nl, http://www.lodebar.nl/pvk/

Eric Backer
Faculty of Information Technology and Systems, Delft University of Technology, Netherlands
e.backer@its.tudelft.nl, http://www-it.et.tudelft.nl/~backer/

In: R. Parncutt, A. Kessler & F. Zimmer (Eds.)
Proceedings of the Conference on Interdisciplinary Musicology (CIM04)
Graz/Austria, 15-18 April, 2004 http://gewi.uni-graz.at/~cim04/

**Background in history of music.** In music history it is often considered important to have a reliable overview of the oeuvre of a certain composer. Unfortunately many pieces exist of which the composer cannot be determined with great certainty. Not rarely this leads to authorship-discussions. Various kinds of evidence are used to defend an attribution. These can be categorized into external and internal evidence (Love, 2002). Stylistic evidence is a subcategory of the latter.

**Background in computing, mathematics and statistics.** In the subdiscipline of machine learning, many algorithms are developed to extract knowledge from measurements in order to make automatic recognition of classes of objects possible. For an overview see e.g. Webb 2002.

**Aims.** The aim of this experiment is to explore the possibilities of machine learning for composer attribution. The focus is on low-level characteristics of counterpoint. So, only polyphonic compositions are taken into account.

**Method** A dataset is made with compositions of five well known eighteenth-century composers. Of all these compositions the composer is known, so they can be used to evaluate the performance of the machine learning tools. Of each composition 20 style markers (*features*) are measured. No widely used and accepted method exists for finding appropriate style markers, so we have to evaluate some. Most of the chosen style markers are low-level counterpoint characteristics. The pattern recognition algorithms are used to obtain knowledge from this data about the uniqueness of each style compared to the others. Also some *classifiers* are built. The algorithms used are: *k-means clustering*, *k-nearest neighbor classifier*, and a *decisiontree* (C4.5). The *fisher-transformation* is used to reduce the dimensionality. In the transformed space, also a nearest neighbor classifier is trained. As estimation of the *true error* rates of these classifiers, the *leave-one-out errors* are computed.

**Results** The clustering shows that the compositions of the each composer do form a cluster in the featurespace. The decisiontree is used to learn which style markers are important for separating the styles. These will appear in the top-nodes. The nearest neighbor classifier performs very well in the transformed featurespace. Very low error-rates can be obtained.

**Conclusions** These results indicate clearly that it is possible to recognize musical style automatically. So, this kind of research can be a valuable addition to more traditional methods of musical style analysis. It offers a quantitative evaluation of the styles rather than the traditional qualitative descriptions.

When listening to the radio, it can happen that a piece of music is broadcasted of which a listener immediately can say: "This must be made by Bach", or "That sounds like the Beatles". This recognition indicates that something in the structure of this music bears the "fingerprint" of its maker. Knowing this, we can ask ourselves if it is possible to develop algorithms with which this kind of attributions can be made automatically. The human pattern recognition system is very complex, and not yet fully understood. But in the past decades many machine learning algorithms have been developed to let computers perform a similar task.

In this paper an experiment is described in which is tried to train a machine to identify the composer of a musical composition, based on a set of *features* (also called *style markers*) extracted from the scores of a set of compositions of which the composer is known. Our aim is both to explore the possibilities of machine learning for this kind of attributions and to learn something about the characteristics of the styles of the composers whose compositions are investigated.

## Non-traditional Authorship Attribution and Stylometry

Because this is a relatively new area of research, it is desirable to find a tradition of research which can be taken as a starting point. Fortunately an area of research exists in which very similar problems are investigated: authorship attribution. Of many historical texts the author is not known, or the attribution made by the source of the text is not trusted. This has caused a lot of research during the entire history of written text. As early as in the fourth century BC, librarians of the famous library of Alexandria studied the "authenticity" of texts attributed to Homer.[1] A project that nowadays, after more than 2000 years, is still not finished. Some other well known examples are the attributions made in the Bible, the plays of Shakespeare and the Federalist Papers.

In the past decades, the ever increasing power of computers made it possible to execute algorithms that demand a lot of computations. The possibilities of this are used in authorship attribution too, which resulted in a research area called *non-traditional authorship attribution*. In this kind of research, it is tried to quantize the style of a certain author. The study of this is called stylometry. It is not obvious what exactly has to be quantized. Many so called style markers are developed. Examples of this are word and sentence lengths, frequencies of certain words and richness of vocabulary. The resulting data is used to classify the texts. Although no widely accepted methodology exists in non-traditional authorship attribution,[2] the results achieved so far, indicate a reasonable probability of success when applying this to music.

The main problem of stylometry is the lack of an underlying theory.[3] Many style markers turn out to be distinctive, but often it is not clear why. In many cases the dimensionality of the data has to be decreased. And if this is done by performing a principal component analysis, it is nearly impossible to understand the meaning of the resulting data. There are, however, some intuitive guidelines which can help with the search for proper style markers. E.g. they have to be independent of the subject written about and of the genre of the text. When counting the number of occurrences of the word "bee" in a text about insects and also in a text about the stock market from the same author, the results will differ, which is not what we want. So, it is better to count context independent words, like "for" and "if". It is also preferable to look for style markers that describe the unconscious processes of text writing. While this kind of style markers are independent of decisions of the author when writing the text, these are expected to have the same values for different texts.

Another problem is the concept *author* itself. In many cases more than one person was involved in the process of creating the text. Editors, for example, can have a profound influence on various characteristics of the text. And, above all, no author works in a vacuum. We have divided the history into different style periods, in which most authors share certain characteristics. So, the question of authorship is not a simple one.

## Music

When trying to attribute a piece of music to a certain composer on stylistic grounds, we have to investigate the score. This is the only entity that connects the activity of the composer with our listening.

Most of the issues that arise when studying authorship of texts, are also of importance when studying authorship of music. For now the question of the selection of style markers is most important. Because there are no generalized rules for determining which style marker will be successful, we will evaluate some which can be expected to be distinctive. A method for (automatic) finding of style markers is desirable, but this is beyond the scope of the present work.

Some studies exist, in which the same problem is encountered, but in most of the cases no rationale is given for the choice of style markers. It is the intuition of the investigator which determines them.

Interesting work has been done by Roger Dannenberg and David Watson. They used machine learning tools to recognize the "mood" of music, such as lyrical, frantic, etc. The goal of this was to build a system in which a human being can improvise together with a computer. They show that it is very well possible to recognize these moods using machine learning tools. Very low error-rates are obtained. Unfortunately they don't mention all the features that were used.

Pedro Ponce de León and José Iñesta used self-organizing neural maps to classify musical styles. Their dataset consists of fragments of jazz and classical music. They give an overview of the features used for this, which involves basic properties of the melodies they extracted from the fragments, such as the number of notes, pitch range, etc. Error-rates between 10% and 20% could be obtained.

Similar quantitative work is also done "by hand". Fred Hofstetter used the Chi-square test to determine whether the difference in interval-characteristics of Eastern- and Western-Europe nineteenth-century string quartets is significant or not. With this he discovered that Eastern melodies had relatively more leaps than Western melodies and that in Eastern melodies relatively more changes of direction can be found.

## Some preliminaries

For this kind of research it is necessary to have the music available in a machine readable format. Unfortunately, a lot of different encodings exist. Each with its own possibilities and limitations. The most widely used representation is MIDI. Because MIDI is a performance-representation and not a representation of the score, it is not suitable for our needs. In MIDI the on- and off-times of pitches are stored. Because a human being who is playing a piece of music, is not very accurate in his timing (and in fact, he should not be), some quantization has to be done before a MIDI-representation can be used for our purpose. When a MIDI-representation is encoded by hand, this drawback is overcome. But, another even more important problem with MIDI is the lack of precision in indicating pitches. There is no difference between e.g. a F sharp and a G flat, because they have the same pitch-number. Since this affects the sizes of most intervals (e.g. an augmented fourth cannot be distinguished from a diminished fifth), determination of harmonic characteristics becomes impossible. There are some extensions of the MIDI-standard in which this limitation is overcome, but these are mutual incompatible and not widely used.

Having a machine readable representation of music is one thing. But then this representation has to be parsed and the contents of the music has to be made available for the routines that will compute the feature values. For this Donncha Ó Maidín (University of Limerick) very kindly made his C++-library CPNView available. This library consists of a number of routines to access the score of a musical composition. CPNView can read the *kern-representation developed by David Huron.[4] This representation is often used for academic research. And a large corpus of encoded music is available from the Center for Computer Assisted Research in the Humanities at Stanford University (CCARH).[5] From this collection a number of compositions is drawn to construct the dataset that is used in the present study.

For the execution of the machine learning algorithms PRTools is used.[6] This is a Matlab-toolbox, created by Bob Duin, in which many pattern recognition algorithms are implemented.

## Dataset

The collection of encoded music at the CCARH consists almost entirely of music from the eighteenth and early nineteenth centuries. Not all of this is suited for our purpose. Many movements from cantatas, oratoria and operas have a basso continuo which is not completely written out. So, some harmonic characteristics cannot be determined. These movements are only used when more than two other voices are active most of the time. In order to obtain reliable feature values, it is also important not to have too short compositions. Furthermore, the compositions must be polyphonic in order to be able to compute all style markers (see below).

With these limitations in mind the following dataset is constructed:[7]

- J.S. Bach: 40 cantata movements.
- J.S. Bach: 33 fugues from "Das wohltemperierte Clavier".
- J.S. Bach: 11 movements from the "Kunst der Fuge".
- J.S. Bach: 8 movements from the violin concertos.
- J.S. Bach: 14 fugues for organ.
- G.F. Handel: 39 movements from the Concerti Grossi op.6.
- G.F. Handel: 14 movements from the trio sonatas op.2 and op.5.

- G.Ph. Telemann: 30 movements from the "Fortsetzung des Harmonischen Gottesdienstes".
- G.Ph. Telemann: 24 movements from the "Musique de table".
- F.J. Haydn: 54 movements from the string quartets.
- W.A. Mozart: 53 movements from the string quartets.

These pieces are chosen from the CCARH-collection such that the variety in the resulting set is optimal, and such that the compositions are to some extend comparable to each other (i.e. not to much variation in the number of voices). Most composers are represented with works in more than one genre. From Mozart and Haydn only string quartet movements are incorporated because it is the most important genre with which they are present in the CCARH-collection.

From this dataset several smaller datasets can be made. All works of one composer can be considered a *class*, but also some composers together can form a class. The following datasets are investigated:

| dataset | classes |
|---|---|
| I | {Bach}, {Telemann}, {Handel}, {Haydn}, {Mozart} |
| II | {Bach}, {Telemann}, {Handel} |
| III | {Bach}, {Telemann, Handel} |
| IV | {Bach}, {Telemann, Handel, Haydn, Mozart} |
| V | {Telemann}, {Handel} |
| VI | {Haydn}, {Mozart} |
| VII | {Telemann, Handel}, {Haydn, Mozart} |

**table 1.** the datasets. Each class is indicated with braces.

With these datasets we can learn something about the differences between the style of J.S. Bach and the styles of the other composers. We can also try to distinguish between composers whose styles are very closely related. Especially the set with Haydn and Mozart will be challenging, since only compositions of the same genre are included.

## Style markers

For each composition in the dataset, the values of 20 style markers are computed. Most of these style markers are low-level properties of counterpoint. When composing polyphonic music, the composer must control the distances between the voices. The way he is doing this, can be expected to be consistent for compositions in different genres and of different dates. The conscious decisions are on higher level. These are e.g. the key, modulations, the development of a theme, the use of certain motifs, etc.

Besides these, some other style markers are evaluated.

**StabTimeslice** The "stability" of the length of the successive timeslices. With a timeslice, the time interval between two changes in the music is meant. The stability is computed by dividing the standard deviation of the lengths of the timeslices by the mean length of the timeslices. This normalization is necessary to compare pieces with different time signatures. So, when having a low value, the music is more like a steady stream, while a larger value indicates more diversity in rhythm.

**DissPart** The fraction of the score in which the sonorities are dissonant. Consonants are: perfect primes, minor and major thirds, perfect fourths, perfect fifths and minor and major sixths. A fourth is considered dissonant if it is between the lowest voice and one of the upper voices. All other intervals are considered dissonant. The total duration of dissonant sonorities is divided by the total duration of the composition.

**BeginBarDiss** The fraction of bars that begins with a dissonant sonority.

**SonorityEntropy** For this style marker, the concept *sonority* is used according to the definition of Robert Mason.[8] In this definition a sonority is a certain type of chord. So e.g. all the major triads are the same sonority, regardless of inversion or pitch. Each sonority has an unique number. For each sonority the total duration of all occurrences in the composition is computed. Then the probabilities of occurrence are estimated using this weighted frequencies. With this probabilities the entropy is computed.

**HarmonyEntropy** The concept *Harmony* is also defined by Mason. It is much like sonority, but now difference is made in pitch. So e.g. a F-major triad and a G-major triad are the same sonority but different harmonies. Again the inversion is not taken into account. The value of this style marker is computed the same way as the SonorityEntropy.

**PitchEntropy** A list of occurrences of all pitches is made. Again the occurrences are weighted by the durations. Of the resulting list, the entropy is computed.

**VoiceDensity** In a polyphonic composition not all voices are active during the whole composition. This style marker indicates the average number of active voices. This is normalized with the total number of voices. For this only bars that are strictly polyphonic are taken into account. I.e. bars in which no voice has more than one note and in which more than one voice is active.

**Intervals** When combining the different voices of a polyphonic composition, the composer has to obey certain constraints. In many of these constraints the vertical distances between the voices are important. This set of style markers measures the amount of some intervals between the different voice-pairs. Systematically all voice-pairs are examined. The total duration of all occurrences of each specific interval is computed and at the end divided by the total duration of all intervals. The intervals are taken modulo one octave. So e.g. a tenth is a third. When the same pitch occurs in more than one voice, it is taken into account once. This is computed for all seconds, all thirds, perfect fourths, augmented fourths, diminished fifths, perfect fifths, all sixths, all sevenths and all octaves.

**Parallels** It can happen that two intervals of the same size succeed each other. This is called a parallel. For this three style markers the amount of parallel thirds, fourths and sixths is computed in the same way as the previous group of style markers. The total duration of all intervals involved in these parallels is added and divided by the total duration of all voicepairs.

**StepSuspension** When a dissonant is sounding between two voices, it often is suspended into a consonant by lowering the lower voice one step. This style marker indicates how many dissonants are suspended this way.

In the following sections these style markers are referred to by their index numbers. They can be found in table 2.

| index | style marker | index | style marker |
|---|---|---|---|
| 1 | StabTimeslice | 11 | PartAgFourths |
| 2 | DissPart | 12 | PartDimFifths |
| 3 | BeginBarDiss | 13 | PartFifths |
| 4 | SonorityEntr. | 14 | PartSixths |
| 5 | HarmonyEntr. | 15 | PartSevenths |
| 6 | PitchEntropy | 16 | PartOctaves |
| 7 | VoiceDensity | 17 | ParThirds |
| 8 | PartSeconds | 18 | ParFourths |
| 9 | PartThirds | 19 | ParSixths |
| 10 | PartFourths | 20 | StepSuspens. |

**Table 2.** The style markers

## Analysis of the data

After computing the values of the style markers, each composition is represented by a vector in a 20-dimensional space, the *featurespace*. Many pattern recognition algorithms operate in this space. They can be used to extract knowledge from the data.

In our dataset, the number of objects per class is far too less to determine what the densities of the classes are like. So, it is better not to use methods that suppose any density. We will use the k-means clustering-algorithm, k-nearest neighbor classifiers and the C4.5 algorithm to build decisiontrees.

For the nearest neighbor classifier and for the k-means algorithm, normalized datasets are used. I.e. for each style marker, the mean is shifted to the origin and all values are divided by the standard deviation. This makes the scales of the style markers more comparable.

It is expected that not all style markers are equally important for discriminating the styles. In most cases it can be better not use them all. Pudils *floating forward selection*[9] algorithm is used to find subsets of style markers. For every dataset, subsets of every possible size are obtained. So for each dataset we obtain 20 sets of style markers.

### Clustering

For all datasets and for all subsets of style markers the k-means clustering-algorithm is executed. A number of so called prototypes are placed in the featurespace. This number of prototypes equals the number of desired clusters. All objects are attributed to the nearest prototype. Then each prototype is moved to the center of its group and the process is repeated. The algorithm ends when the prototypes remain at the same positions. The algorithm is initiated with a number of prototypes that equals the number of classes in the dataset. The prototypes are placed at the known class means.

In table 3 some results of the clustering are shown. For each dataset the set of style markers is shown with which low confusion between the known classes and the discovered clusters is obtained. And for which as less as possible style markers are used.

| set | style markers | errors |
|---|---|---|
| I | 1, 2, 5, 8, 9, 10, 13, 14, 17 | 110 (35.9%) |
| II | 2, 14, 17 | 30 (15.1%) |
| III | 1, 2, 4, 13, 17 | 13 (6.3%) |
| IV | 1, 2, 17 | 39 (12.3%) |
| V | 4, 10, 14 | 11 (10.3%) |
| VI | 13 | 37 (34.6%) |
| VII | 13, 18 | 42 (19.6%) |

**Table 3.** For each dataset the error made when clustering. Also the used style markers are shown.

These results indicate that the various classes really do form coherent structures. Except for the first dataset, it appears that all classes can be found with just a few style markers. It is obvious that separating the string quartets of Mozart and Haydn is the most difficult task. Because the errors on datasets II and VII are relatively low, the large error for dataset I is also caused by this. So within the baroque group and between the classicism and baroque groups separation is very well possible, and within the classicism group, separation is more difficult.

### Nearest neighbor classification

A nearest neighbor classifier assigns an unknown object to the class of the known object that is nearest in the featurespace. It is also possible to use more than one neighbor. In this case, a majority vote is taken to determine the class of the unknown object.

For our purpose, we ask the following question. If we want to classify an unknown composition with the available data, which style makers and how many neighbors do we need? I.e. which classifier has the lowest error rate?

In order to obtain a reliable estimation of the true error of the classifier and to avoid overfitting, train- and testsets have to be made for training and testing the classifiers. The size of these sets has to be chosen such that a reliable indication of the true error of the classifier is obtained.

If we test the classifier with *hold out validation*, not all objects will be used for learning. This results in a very pessimistic estimate of the true error. Therefore it is better to use more advanced methods to test the classifier. Since we do not have the luxury of having much data, *leave-one-out validation* is preferable. One object is isolated. On all other objects a classifier is trained. Then the one isolated object is used to test that classifier. This is done for each object in the dataset. The error rate is the fraction of objects that is misclassified. This method reduces the bias in the estimation of the true error and increases the variance. But, since our datasets are not very small, the effect of the increased variance will not dominate the results.

Table 4 shows some results. For each dataset a classifier with a low error rate is described. These classifiers are characterized by the used style markers and by the number of neighbors. Not always the classifier with the lowest error rate is given. If the second best has a much lower number of neighbors or a much lower number of style markers, the second best classifier is shown.

| Set | *k* | Style markers | loo err |
|---|---|---|---|
| I | 11 | 1, 2, 5, 6, 7, 8, 9, 10, 11, 13, 14, 17, 19, 20 | 0.265 |
| II | 13 | 1, 2, 6, 10, 13, 14, 17 | 0.096 |
| III | 7 | 1, 2, 17 | 0.053 |
| IV | 5 | 1, 2, 17 | 0.066 |
| V | 7 | 3, 4, 7, 8, 10, 11, 12, 14, 19, 20 | 0.084 |
| VI | 3 | 1, 6, 7, 11, 13 | 0.243 |
| VII | 11 | 2, 5, 7, 8, 11, 13, 16, 19 | 0.089 |

**Table 4.** Some classifiers with low error rates. *k* is the number of neighbors

When looking to the error-rates, we can see the same pattern as with the results of the k-means algorithm. Datasets I and VI are the most difficult to classify due to the presence of the compositions of Haydn and Mozart. It can also be seen that the chosen style markers are roughly the same as the style markers which were best for clustering.

With a so called *fisher transformation* we can map the objects into a lower dimensional space. This new space is chosen such that the separability between the classes is optimal. I.e. axes that span the new space are chosen such that the scatter within the classes is minimized and the scatter between the classes is maximized. This is done for all datasets. Also in the transformed space nearest neighbor classifiers are evaluated for different numbers of neighbors. The results are in table 5.

| Dataset | *k* | loo-error |
|---|---|---|
| I | 15 | 0.1993 |
| II | 17 | 0.0704 |
| III | 15 | 0.0481 |
| IV | 15 | 0.0599 |
| V | 9 | 0.0841 |
| VI | 7 | 0.2056 |
| VII | 11 | 0.0654 |

**Table 5.** the leave-one-out error rates of the k-nearest neighbor classifiers in the transformed featurespade

When comparing these results to the results on the untransformed dataset, one can see that this transformation reduces the errors. So in order to build good performing classifiers, this transformation is very useful. But since the new featurespace doesn't relate directly to the style markers, it doesn't give much more insight in the data.

When removing three compositions of Bach from dataset III even a lower error rate can be obtained (2.4%). The removed compositions BWV 4.5 and 550 are youth works. The other removed composition BWV 6.3 has only 2 active voices most of the time. One of them is a chorale melody. These compositions can be considered outliers, so it is not a violation to the dataset to remove them.

## Decisiontrees

For each of the datasets a decisiontree is built. Since the C4.5 algorithm is not deterministic, the best of 50 trials is chosen. I.e. the one with the lowest apparent error. These trees give insight in the relevance of the various style markers. The style marker that is most discriminative will appear in the top node of the tree. Style markers that are low in the tree, or that are not in the tree at all, can be considered irrelevant for separating the classes.

From the tree that is built for dataset I, we can learn that all compositions of Telemann, and 40 of Handel are in the subspace with a low amount of dissonants and a low amount of augmented fourths. In the same subspace 8 compositions of Haydn, 17 of Mozart and 5 of Bach can be found. So, it can be stated that the style of Handel and Telemann is characterized by consonant sonorities and a low amount of augmented fourths.

73 of the 92 compositions of Bach are in the subspace that is shown in figure 1. Together with 4 compositions of Handel, 2 of Mozart and 2 of Haydn. For those who are familiar with the music of Bach, this is not a surprise. Bach wrote more dissonant sonorities than his contemporaries. But in the generations that follow him, boundaries were shifting and composers allowed themselves more freedom in this. So the DissPart style marker separates Handel and Telemann from Bach. But, it doesn't separate Bach from Haydn and Mozart. For this, other style markers are needed. The parallel third is a device to enrich the sonorities, but does not add very much to the musical contents of a composition. The amount of parallel thirds is low in Bach's music. Probably he wouldn't be satisfied by a lot of parallel thirds. The rhythmic stability is also a characteristic of Bach's music, which in many cases is like a steady stream. This is especially the case in the polyphonic compositions, which are taken into account here. The combination of these three characteristics leads to the style of Bach.

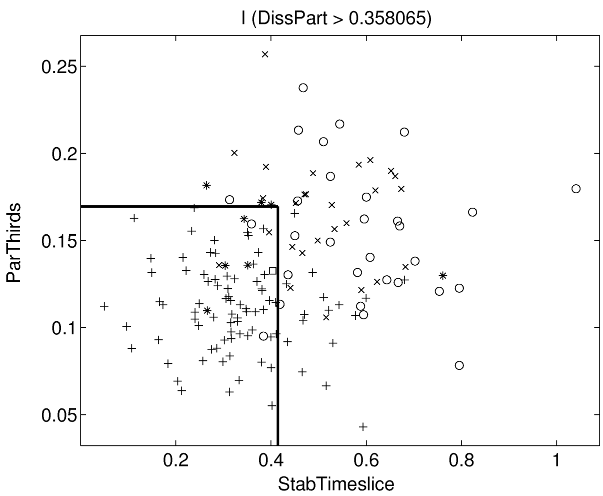


**Figure 1.** Scatterplot with the features that characterize Bach's style (+). Compositions with DissPart ≤ 0.358065 are not shown.

The compositions of Haydn and Mozart are scattered all over the tree. There is no subspace which is clearly the domain of these two composers. In order to learn more about the styles of these composers, it is better to examine the tree built from dataset VI.

Unfortunately this tree is rather deep, which is not surprising because we already saw that it is difficult to separate the compositions of these two composers. Making a highly pruned tree is more insightful. This is done by setting the maximum number of objects at a leaf to eight. In the resulting tree, the voicedensity is the most important style marker. The densities of the two classes are shown in figure 2. Roughly spoken, in Mozart's string quartets voices are more active than in Haydn's. But, as can be seen, there is much overlap. In the subspace with low voicedensities, 32 movements of Haydn and 11 of Mozart can be found, while the complementing subspace is occupied by 22 movements of Haydn and 42 of Mozart.

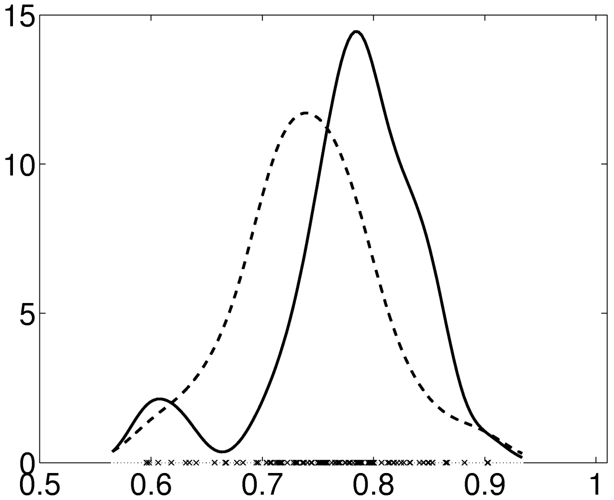


**Figure 2.** The densities of the voicedensity for Haydn (dashed) and Mozart (solid)

To learn what separates the styles of Handel and Telemann we can inspect the tree made from dataset V. From this tree it is clear that the amount of fourths and sixths are the most discriminative style markers. 39 of the 53 compositions of Handel are in the subspace with high amount of fourths and high amount of sixths, while most of Telemann's compositions are in the combination of two subspaces. Roughly spoken, Telemann's compositions are characterized by a low amount of fourths and sixths, and when the amount of fourths is higher, the amount of sixths is lower. When making a scatterplot (figure 3), we can understand why the tree is made this way

# Conclusions

The obtained results indicate clearly that it is possible to recognize musical style automatically. So, this kind of research can be a valuable addition to more traditional methods of musical style analysis. It offers a quantitative evaluation of the styles rather than the traditional qualitative descriptions. It is important not to see this as a replacement, but as an addition. Combining results from different viewpoints, will give more robust knowledge. The results of this study are a promise for the future, in which we can expect further increase in the computational power as well as further increase in the understanding and application of pattern recognition techniques.

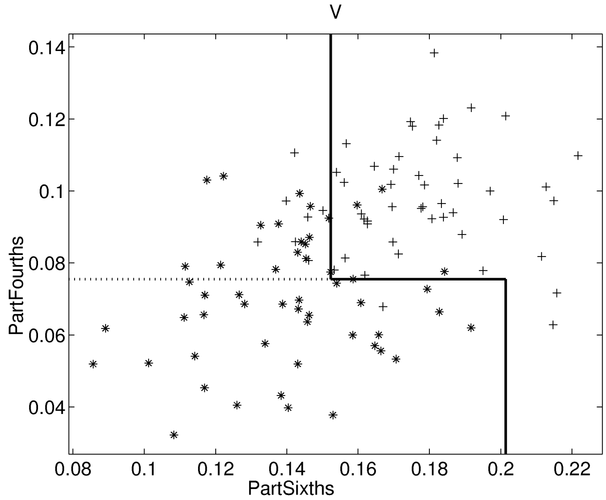


**Figure 3.** scatterplot of the features PartFourths and PartSixths of classes Haendel (+) and Telemann (*), with the discriminant of the decisiontree.

This also means that this kind of research can be helpful in authorship disputes. It reveals characteristics of the compositions that cannot be known from normal perception. And, therefore, also not from the perception of the composers. When one composer imitates the style of another, it is not likely that he has much control on the characteristics that are examined in this study. Compositions which are very like each other for a listener, can be located in different places in the featurespace.

With the results of the k-means clustering-algorithm it is shown that the compositions of the various composers do have tendency to cluster. This is an indication that the chosen style markers are suited for separating the styles represented in the dataset.

It is shown that projection of the data on the fisher-discriminants leads to more accurate attributions. The resulting featurespace is difficult to understand from a music-theory point of view, but when using this in authorship problems, this mapping can be made to get better results.

By building decisiontrees, we can learn something about the styles of the different composers. They all had their own preferences. It is shown that with these style markers Bach's style can be distinguished from his contemporaries with great accuracy. The style marker that is most important for this is the amount of dissonant sonorities. The combination of a high amount of dissonant sonorities, a low amount of parallel thirds and a steady rhythm leads to the style of Bach.

Although their styles are closely related, there is not much confusion between the compositions of Handel and Telemann. So the used style markers are also suited to uncover the differences between the styles of these two composers. It turns out that Handel wrote more fourths and sixths than Telemann.

There is much more confusion between Haydn and Mozart. This could be expected since all compositions considered are in the same genre. And the composers have influenced each other. The style marker that turns out to be the most important is the voicedensity. Mozart wrote thicker textures than Haydn. But, to get a better description of their styles other style markers may be necessary.

## Future work

The present work indicates the possibilities of success when using these pattern recognition techniques for more specific problems. It can be used to shed new light on unsolved authorship problems.

It is also possible to follow development of style over a certain period of time. What separates the seventeenth form the eighteenth century? How does the individual style of a composer develop?

Of course, the set of style markers can be extended. E.g. addition of style markers that describe the melodic characteristics (intervals, step vs. leap, etc.), style markers that describe harmonic properties, or more detailed contrapuntal properties. There are many more possibilities, but maybe it is more important to find a more fundamental approach for finding style markers. There is a need for an underlying theory. It might be interesting to develop psychological models for the act of composing. This could give us an indication of which style makers reflects the unconscious processes in composing music.

It is also interesting to think about what is necessary to describe a musical style entirely. Probably this is not possible, but maybe something can be said about the completeness of the set of style markers.

**Acknowledgements** We wish to thank Frans Wiering (University of Utrecht) for pointing us out to CIM04 and for his advice.

## References

Dannenberg, T., and Watson, "A Machine Learning Approach to Musical Style Recognition", *Proceedings of the 1997 International Computer Music Conference*, pp. 344-347.

Jain, A.,and Zongker, D., "Feature Selection: Evaluation, Application, and Small Sample Performance", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1997), 153-158.

Love, H., *Attributing Authorship: An Introduction*, Cambridge, 2002.

Mason, R.M., *Modern Methods of Music Analysis using Computers*, Peterborough, 1985.

Ponce de León, P.J., J.M. Iñesta, "Feature-driven recognition of music styles", *Lecture Notes in Computer Science*, 2652 (2003), pp. 773-781.

Rudman, J., *The Hypothetical and Theoretical Underpinnings of Non-traditional Authorship Attribution Studies: Assumptions, Presumptions, and Verifiable Constructs.*, http://www.iath.virginia.edu/ach-allc.99/proceedings/rudman.html.

Webb, A., *Statistical Pattern Recognition*, Chichester, 2002.

---

[1] Love (2002), p.15

[2] Rudman (1999) and Love (2002), p.156vv

[3] Love (2002), p.156vv

[4] http://dactyl.som.ohio-state.edu/Humdrum/representations/kern.html

[5] http://www.ccarh.org

[6] http://www.ph.tn.tudelft.nl/~bob/PRTOOLS.html

[7] On http://www.musical-style-recognition.net a complete list of all compositions can be found.

[8] Mason (1985), p.21

[9] evaluated in Jain (1997)