Single Channel Music Sound Separation Based on Spectrogram Decomposition and Note Classification

Wenwu Wang* and Hafiz Mustafa

Centre for Vision, Speech and Signal Processing (CVSSP)
University of Surrey, GU2 7XH, UK
{w.wang,hm00045}@surrey.ac.uk
http://www.surrey.ac.uk/cvssp

Abstract. Separating multiple music sources from a single channel mixture is a challenging problem. We present a new approach to this problem based on non-negative matrix factorization (NMF) and note classification, assuming that the instruments used to play the sound signals are known a priori. The spectrogram of the mixture signal is first decomposed into building components (musical notes) using an NMF algorithm. The Mel frequency cepstrum coefficients (MFCCs) of both the decomposed components and the signals in the training dataset are extracted. The mean squared errors (MSEs) between the MFCC feature space of the decomposed music component and those of the training signals are used as the similarity measures for the decomposed music notes. The notes are then labelled to the corresponding type of instruments by the K nearest neighbors (K-NN) classification algorithm based on the MSEs. Finally, the source signals are reconstructed from the classified notes and the weighting matrices obtained from the NMF algorithm. Simulations are provided to show the performance of the proposed system.

Keywords: Non-negative matrix factorization, single-channel sound separation, Mel frequency cepstrum coefficients, instrument classification, K nearest neighbors, unsupervised learning.

1 Introduction

Recovering multiple unknown sources from a one-microphone signal, which is an observed mixture of these sources, is referred to as the problem of single-channel (or monaural) sound source separation. The single-channel problem is an extreme case of under-determined separation problems, which are inherently ill-posed, i.e., more unknown variables than the number of equations. To solve the problem, additional assumptions (or constraints) about the sources or the propagating channels are necessary. For an underdetermined system with two

* The work of W. Wang was supported in part by an Academic Fellowship of the RCUK/EPSRC (Grant number: EP/C509307/1).

S. Ystad et al. (Eds.): CMMR 2010, LNCS 6684, pp. 84–101, 2011.

microphone recordings, it is possible to separate the sources based on spatial diversity using determined independent component analysis (ICA) algorithms and an iterative procedure [17]. However, unlike the techniques in e.g. ADRess [2] and DUET [18] that require at least two mixtures, the cues resulting from the sensor diversity are not available in the single channel case, and thus separation is difficult to achieve based on ICA algorithms.

Due to the demand from several applications such as audio coding, music information retrieval, music editing and digital library, this problem has attracted increasing research interest in recent years [14]. A number of methods have been proposed to tackle this problem. According to the recent review by Li et al. [14], these methods can be approximately divided into three categories: (1) signal modelling based on traditional signal processing techniques, such as sinusoidal modelling of the sources, e.g. [6], [23], [24]; (2) learning techniques based on statistical tools, such as independent subspace analysis [4] and non-negative matrix (or tensor) factorization, e.g. [19], [20], [27], [28], [25], [8], [30]; (3) psychoacoustical mechanism of human auditory perception, such as computational auditory scene analysis (CASA), e.g. [15], [3], [26], [32], [14]. Sinusoidal modelling methods try to decompose the signal into a combination of sinusoids, and then estimate their parameters (frequencies, amplitudes, and phases) from the mixture. These methods have been used particularly for harmonic sounds. The learning based techniques do not exploit explicitly the harmonic structure of the signals, instead they use the statistical information that is estimated from the data, such as the independence or sparsity of the separated components. The CASA based techniques build separation systems on the basis of the perceptual theory by exploiting the psychoacoustical cues that can be computed from the mixture, such as common amplitude modulation.

In this paper, a new algorithm is proposed for the problem of single-channel music source separation. The algorithm is based mainly on the combination of note decomposition with note classification. The note decomposition is achieved by a non-negative matrix factorization (NMF) algorithm. NMF has been previously used for music sound separation and transcription, see e.g. [11], [1], [7], [20], [29], [30]. In this work, we first use the NMF algorithm in [25] to decompose the spectrogram of the music mixture into building components (musical notes). Then, Mel Frequency Cepstrum Coefficients (MFCCs) feature vectors are extracted from the segmented frames of each decomposed note. To divide the separated notes into their corresponding instrument categories, the K nearest neighbor (NN) classifier [10] is used. The K-NN classifier is an algorithm that is simple to implement and also provides good classification performance. The source signals are reconstructed by combining the notes having same class labels. The remainder of the paper is organized as follows. The proposed separation system is described in Section 2 in detail. Some preliminary experimental results are shown in Section 3. Discussions about the proposed method are given in Section 4. Finally, Section 5 summarises the paper.

2 The Proposed Separation System

This section describes the details of the processes in our proposed sound source separation system. First, the single-channel mixture of music sources is decomposed into basic building blocks (musical notes) by applying the NMF algorithm. The NMF algorithm describes the mixture in the form of basis functions and their corresponding weights (coefficients) which represent the strength of each basis function in the mixture. The next step is to extract the feature vectors of the musical notes and then classify the notes into different source streams. Finally, the source signals are reconstructed by combining the notes with the same class labels. In this work, we assume that the instruments used to generate the music sources are known a priori. In particular, two kinds of instruments, i.e. piano and violin, were used in our study. The block diagram of our proposed system is depicted in Figure 1.

2.1 Music Decomposition by NMF

In many data analysis tasks, it is a fundamental problem to find a suitable representation of the data so that the underlying hidden structure of the data may be revealed or displayed explicitly. NMF is a data-adaptive linear representation technique for 2-D matrices that was shown to have such potentials. Given a non-negative data matrix $\mathbf{X}$, the objective of NMF is to find two non-negative matrices $\mathbf{W}$ and $\mathbf{H}$ [12], such that

$$\mathbf{X} = \mathbf{WH} \tag{1}$$

In this work, $\mathbf{X}$ is an $S \times T$ matrix representing the mixture signal, $\mathbf{W}$ is the basis matrix of dimension $S \times R$, and $\mathbf{H}$ is the weighting coefficient matrix of

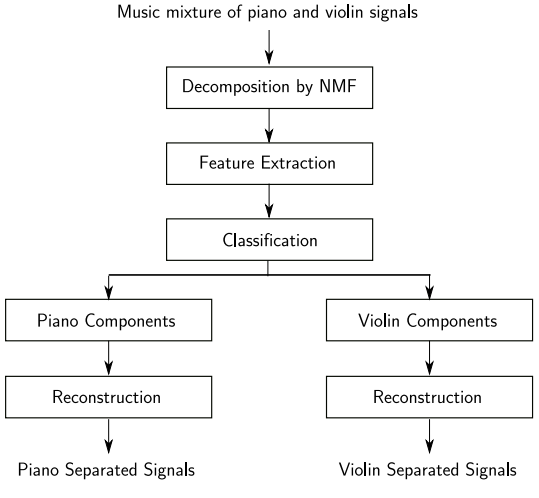

Fig. 1. Block diagram of the proposed system

dimension $R \times T$. The number of bases used to represent the original matrix is described by R, i.e. the decomposition rank. Due to non-negativity constraints, this representation is purely additive. Many algorithms can be used to find the suitable pair of $\mathbf{W}$ and $\mathbf{H}$ such that the error of the approximation is minimised, see e.g. [12], [13], [7], [20] and [30]. In this work, we use the algorithm proposed in [25] for the note decomposition. In comparison to the classical algorithm in [12], this algorithm considers additional constraints from the structure of the signal.

Due to the non-negativity constraints, the time-domain signal (with negative values) needs to be transformed into another domain so that only non-negative values are present in $\mathbf{X}$ for an NMF algorithm to be applied. In this work, the music sound is transformed into the frequency domain using, e.g. the short-time Fourier transform (STFT). The matrix $\mathbf{X}$ is generated as the spectrogram of the signal, and in our study, the frame size of each segment equals to 40 ms, and 50 percent overlaps between the adjacent frames are used. An example of matrix $\mathbf{X}$ generated from music signals is shown in Figure 2, where two music sources with each having a music note repeating twice were mixed together. One of the sources contains musical note G4, and the other is composed of note A3. The idea of decomposing the mixture signal into individual music components is based on the observation that a music signal may be represented by a set of basic building blocks such as musical notes or other general harmonic structures. The basic building blocks are also known as basis vectors and the decomposition of the single-channel mixture into basis vectors is the first step towards the separation of multiple source signals from the single-channel mixture. If different sources in the mixture represent different basis vectors, then the separation problem can be regarded as a problem of classification of basis vectors into different categories. The source signals can be obtained by combining the basis vectors in each category.

The above mixture (or NMF) model can be equally written as

$$\mathbf{X} = \sum_{r=1}^{R} \mathbf{w}_r \mathbf{h}_r \tag{2}$$

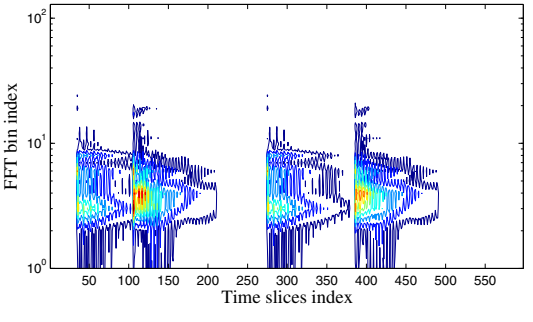

Fig. 2. The contour plot of a sound mixture (i.e. the matrix $\mathbf{X}$) containing two different musical notes G4 and A3

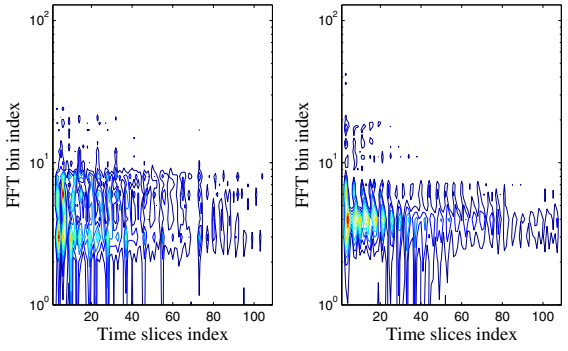

Fig. 3. The contour plots of the individual musical notes which were obtained by applying an NMF algorithm to the sound mixture $\mathbf{X}$. The separated notes G4 and A3 are shown in the left and right plot respectively.

where $\mathbf{w}_r$ is the r^{th} column of $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_R]$ which contains the basis vectors, and $\mathbf{h}_r$ is the r^{th} row of $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_R]^T$ which contains the weights or coefficients of each basis function in matrix $\mathbf{W}$, where the superscript T is a matrix transpose. Many algorithms including those mentioned above can be applied to obtain such basis functions and weighting coefficients. For example, using the algorithm developed in [30], we can decompose the mixture in Figure 2, and the resulting basis vectors (i.e. the decomposed notes) are shown in Figure 3. From this figure, it can be observed that both note G4 and A3 are successfully separated from the mixture.

As a prior knowledge, given the mixture of musical sounds containing two sources (e.g. piano and violin), two different types of basis functions are learnt from the decomposition by the NMF algorithm. The magnitude spectrograms of the basis components (notes) of the two different sources in the mixture are obtained by multiplying the columns of the basis matrix $\mathbf{W}$ to the corresponding rows of the weight matrix $\mathbf{H}$. The columns of matrix $\mathbf{W}$ contain the information of musical notes in the mixture and corresponding rows of matrix $\mathbf{H}$ describe the strength of these notes. Some rows in $\mathbf{H}$ do not contain useful information and are therefore considered as noise. The noise components are considered separately in the classification process to improve the quality of the separated sources.

2.2 Feature Extraction

Feature extraction is a special form of dimensionality reduction by transforming the high dimensional data into a lower dimensional feature space. It is used in both the training and classification processes in our proposed system. The audio features that we used in this work are the MFCCs. The MFCCs are extracted on a frame-by-frame basis. In the training process, the MFCCs are extracted from a training database, and the feature vectors are then formed from these coefficients. In the classification stage, the MFCCs are extracted similarly from

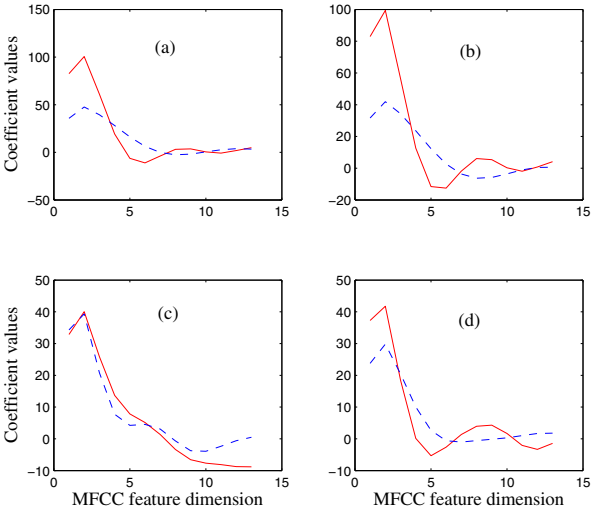

Fig. 4. The 13-dimensional MFCC feature vectors calculated from two selected frames of the four audio signals: (a) "Piano.ff.A0.wav", (b) "Piano.ff.B0.wav", (c) "Violin.pizz.mf.sulG.C4B4.wav", and (d) "Violin.pizz.pp.sulG.C4B4.wav". In each of the four plots, the solid and dashed lines represent the two frames (i.e. the 400th and 900th frame), respectively.

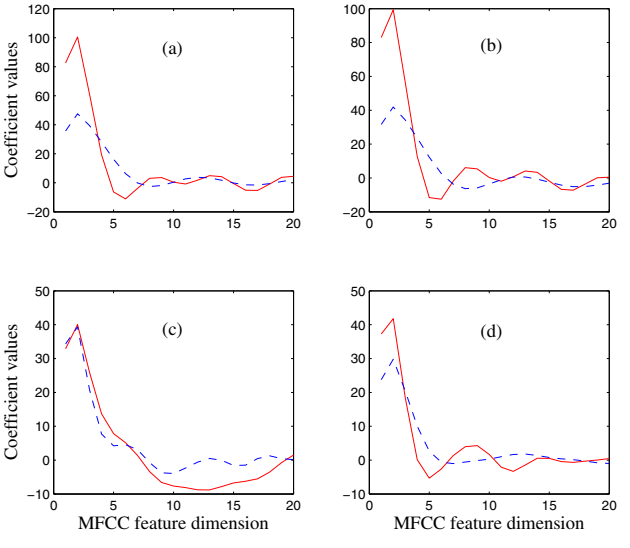

Fig. 5. The 20-dimensional MFCC feature vectors calculated from two selected frames of the four audio signals: (a) "Piano.ff.A0.wav", (b) "Piano.ff.B0.wav", (c) "Violin.pizz.mf.sulG.C4B4.wav", and (d) "Violin.pizz.pp.sulG.C4B4.wav". In each of the four plots, the solid and dashed lines represent the two frames (i.e. the 400th and 900th frame), respectively.

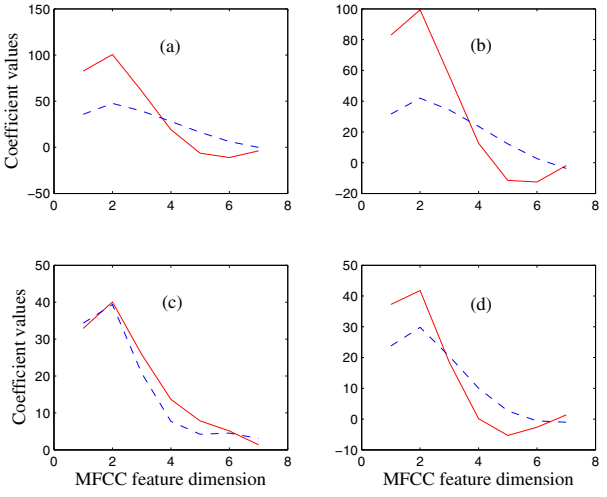

Fig. 6. The 7-dimensional MFCC feature vectors calculated from two selected frames of the four audio signals: (a) "Piano.ff.A0.wav", (b) "Piano.ff.B0.wav", (c) "Violin.pizz.mf.sulG.C4B4.wav", and (d) "Violin.pizz.pp.sulG.C4B4.wav". In each of the four plots, the solid and dashed lines represent the two frames (i.e. the 400th and 900th frame), respectively.

the decomposed notes obtained by the NMF algorithm. In our experiments, the frame size of 40 ms is used, which equals to 1764 samples when the sampling frequency is 44100 Hz. Examples of such feature vectors are shown in Figure 4, where the four audio files ("Piano.ff.A0.wav", "Piano.ff.B0.wav", "Violin.pizz.mf.sulG.C4B4.wav", and "Violin.pizz.pp.sulG.C4B4.wav") were chosen from the The University of Iowa Musical Instrument Samples Database [21] and the feature vectors are 13-dimensional. Different dimensional features have also been examined in this work. Figure 5 and 6 show the 20-dimensional and 7-dimensional MFCC feature vectors computed from the same audio frames and from the same audio signals as those in Figure 4. In comparison to Figure 4, it can be observed that the feature vectors in Figure 5 and 6 have similar shapes, even though that the higher dimensional feature vectors show more details about the signal. However, it inevitably incurs a higher computational cost if the feature dimension is increased. In our study, we choose to compute a 13 dimensional MFCCs vector for each frame in the experiments, which offers a good trade-off between the classification performance and the computational efficiency.

2.3 Classification of Musical Notes

The main objective of classification is to maximally extract patterns on the basis of some conditions and is to separate one class from another. The K-NN classifier, which uses a classification rule without having the knowledge of the distribution of measurements in different classes, is used in this paper for the separation

Table 1. The musical note classification algorithm

1) Calculate the 13-D MFCCs feature vectors of all the musical examples in the training database with class labels. This creates a feature space for the training data.
2) Extract similarly the MFCCs feature vectors of all separated components whose class labels need to be determined.
3) Assign the labels to all the feature vectors in the separated components to the appropriate classes via the K-NN algorithm.
4) The majority vote of feature vectors determines the class label of the separated components.
5) Optimize the classification results by different choices of K.

of piano and violin notes. The basic steps in music note classification include preprocessing, feature extraction or selection, classifier design and optimization. The main steps used in our system are detailed in Table 1.

The main disadvantage of the classification technique based on simple "majority voting" is that the classes with more frequent examples tend to come up in the K-nearest neighbors when the neighbors are computed from a large number of training examples [5]. Therefore, the class with more frequent training examples tends to dominate the prediction of the new vector. One possible technique to solve this problem is to weight the classification based on the distance from the test pattern to all of its K nearest neighbors.

2.4 K-NN Classifier

This section briefly describes the K-NN classifier used in our algorithm. K-NN is a simple technique for pattern classification and is particularly important for non-parametric distributions. The K-NN classifier labels an unknown pattern x by the majority vote of its K-nearest neighbors [5], [9]. The K-NN classifier belongs to a class of techniques based on non-parametric probability density estimation. Suppose, there is a need to estimate the density function $P(x)$ from a given dataset. In our case, each signal in the dataset is segmented to 999 frames, and a feature vector of 13 MFCC coefficients is computed for each frame. Therefore, the total number of examples in the training dataset is 52947. Similarly, an unknown pattern x is also a 13 dimensional MFCCs feature vector whose label needs to be determined based on the majority vote of the nearest neighbors. The volume V around an unknown pattern x is selected such that the number of nearest neighbors (training examples) within V are 30. We are dealing with the two-class problem with prior probability $P(\omega_i)$. The measurement distribution of the patterns in class ω_i is denoted by $P(x \mid \omega_i)$. The measurement of posteriori class probability $P(\omega_i \mid x)$ decides the label of an unknown feature vector of the separated note. The approximation of $P(x)$ is given by the relation [5], [10]

$$P(x) \simeq \frac{K}{NV} \tag{3}$$

where N is the total number of examples in the dataset, V is the volume surrounding unknown pattern x and K is the number of examples within V. The class prior probability depends on the number of examples in the dataset

$$P(\omega_i) = \frac{N_i}{N} \tag{4}$$

and the mesurement distribution of patterns in class ω_i is defined as

$$P(x \mid \omega_i) = \frac{K_i}{N_i V} \tag{5}$$

According to the Bayes theorem, the posteriori probability becomes

$$P(\omega_i \mid x) = \frac{P(x \mid \omega_i) P(\omega_i)}{P(x)} \tag{6}$$

Based on the above equations, we have [10]

$$P(\omega_i \mid x) = \frac{K_i}{K} \tag{7}$$

The discriminant function $g_i(x) = \frac{K_i}{K}$ assigns the class label to an unknown pattern x based on the majority of examples K_i of class ω_i in volume V.

2.5 Parameter Selection

The most important parameter in the K-NN algorithm is the user-defined constant K. The best value of K depends upon the given data for classification [5]. In general, the effect of noise on classification may be reduced by selecting a higher value of K. The problem arises when a large value of K is used for less distinct boundaries between classes [31]. To select good value of K, many heuristic techniques such as cross-validation may be used. In the presence of noisy or irrelevant features the performance of K-NN classifier may degrade severely [5]. The selection of feature scales according to their importance is another important issue. For the improvement of classification results, a lot of effort has been devoted to the selection or scaling of the features in a best possible way. The optimal classification results are achieved for most datasets by selecting $K = 10$ or more.

2.6 Data Preparation

For the classification of separated components from mixture, the features i.e. the MFCCs, are extracted from all the signals in the training dataset and put the label on all feature vectors according to their classes (piano or violin). The labels of the feature vectors of the separated components are not known which need to be classified. Each feature vector consist of 13 MFCCs. When computing the MFCCs, the training signals and the separated components are all divided into frames with each having a length of 40 ms and 50 percent overlap between

the frames is used to avoid discontinuities between the neighboring frames. The similarity measure of the feature vectors of the separated components to the feature vectors obtained from the training process determines which class the separated notes belong to. This is achieved by the K-NN classifier. If majority vote goes to the piano, then a piano label is assigned to the separated component and vice-versa.

2.7 Phase Generation and Source Reconstruction

The factorization of magnitude spectrogram by the NMF algorithm provides frequency-domain basis functions. Therefore, the reconstruction of source signals from the frequency-domain bases is used in this paper, where the phase information is required. Several phase generation methods have been suggested in the literature. When the components do not overlap each other significantly in time and frequency, the phases of the original mixture spectrogram produce good synthesis quality [23]. In the mixture of piano and violin signals, significant overlapping occurs between musical notes in the time domain but the degree of overlapping is relatively low in the frequency domain. Based on this observation, the phases of the original mixture spectrogram are used to reconstruct the source signals in this work. The reconstruction process can be summarised briefly as follows. First, the phase information is added to each classified component to obtain its complex spectrum. Then the classified components from the above sections are combined to the individual source streams, and finally the inverse discrete Fourier Transform (IDFT) and the overlap-and-add technique are applied to obtain the time-domain signal. When the magnitude spectra are used as the basis functions, the frame-wise spectra are obtained as the product of the basis function with its gain. If the power spectra are used, a square root needs to be taken. If the frequency resolution is non-linear, additional processing is required for the re-synthesis using the IDFT.

3 Evaluations

Two music sources (played by two different instruments, i.e. piano and violin) with different number of notes overlapping each other in the time domain, were used to generate artificially an instantaneous mixture signal. The lengths of piano and violin source signals are both 20 seconds, containing 6 and 5 notes respectively. The K-NN classifier constant K was selected as $K = 30$. The signal-to-noise ratio (SNR), defined as follows, was used to measure the quality of both the separated notes and the whole source signal,

$$SNR(m, j) = \frac{\sum_{s,t} [\mathbf{X}_m]_{s,t}^2}{\sum_{s,t} ([\mathbf{X}_m]_{s,t} - [\mathbf{X}_j]_{s,t})^2} \tag{8}$$

where s and t are the row and column indices of the matrix respectively. The SNR was computed based on the magnitude spectrograms $\mathbf{X}_m$ and $\mathbf{X}_j$ of the m^{th} reference and the j^{th} separated component to prevent the reconstruction

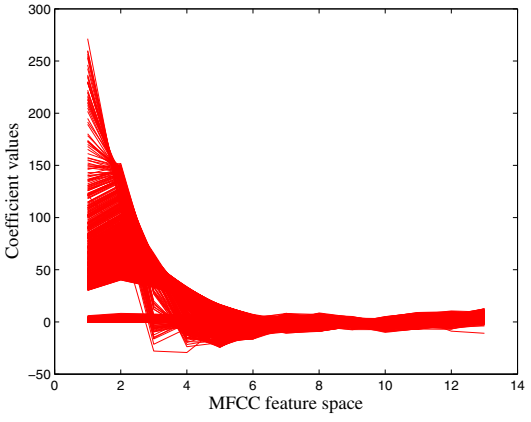

Fig. 7. The collection of the audio features from a typical piano signal (i.e. "Piano.ff.A0.wav") in the training process. In total, 999 frames of features were computed.

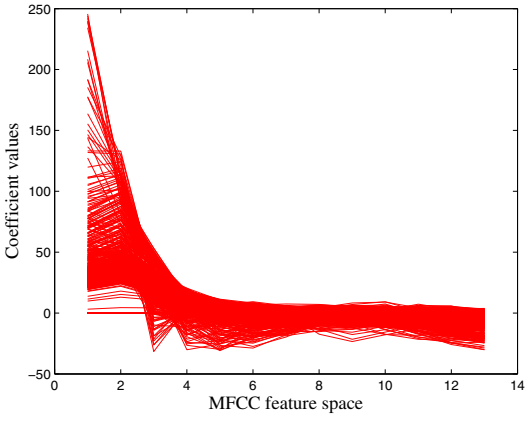

Fig. 8. The collection of the audio features from a typical violin signal (i.e. "Violin.pizz.pp.sulG.C4B4.wav") in the training process. In total, 999 frames of features were computed.

process from affecting the quality [22]. For the same note, $j = m$. In general, higher SNR values represent better separation quality of the separated notes and source signals, vice-versa. The training database used in the classification process was provided by the McGill University Master Samples Collection [16], University of Iowa website [21]. It contains 53 music signals with 29 of which are piano signals and the rest are violin signals. All the signals were sampled at 44100 Hz. The reference source signals were stored for the measurement of separation quality.

For the purpose of training, the signals were firstly segmented into frames, and then the MFCC feature vectors were computed from these frames. In total,

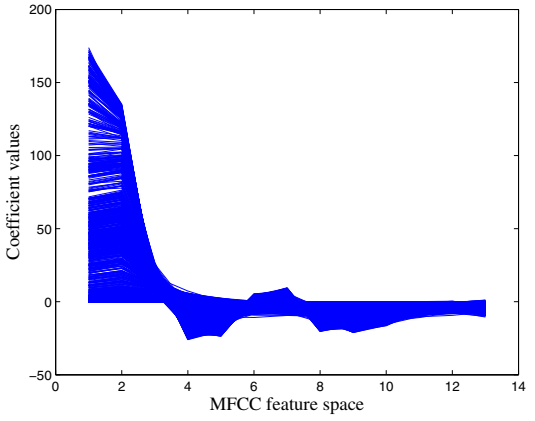

Fig. 9. The collection of the audio features from a separated speech component in the testing process. Similar to the training process, 999 frames of features were computed.

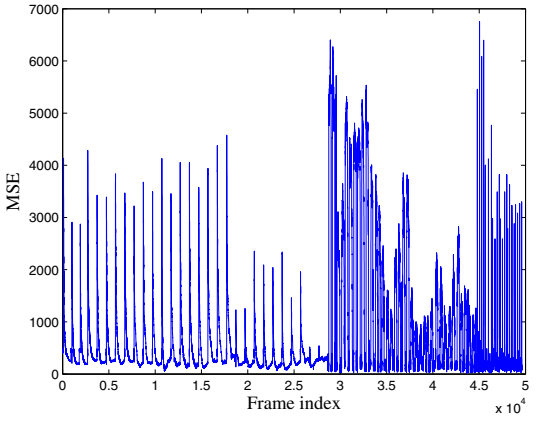

Fig. 10. The MSEs between the feature vector of a frame of the music component to be classified and those from the training data. The frame indices in the horizontal axis are ranked from the lower to the higher. The frame index 28971 is the highest frame number of the piano signals. Therefore, on this plot, to the left of this frame are those from piano signals, and to the right are those from the violin signals.

999 frames were computed for each signal. Figures 7 and 8 show the collection of the features from the typical piano and violin signals (i.e. "Piano.ff.A0.wav" and "Violin.pizz.pp.sulG.C4B4.wav") respectively. In both figures, it can be seen that there exist features whose coefficients are all zeros due to the silence part of the signals. Before running the training algorithm, we performed feature selection by removing such frames of features. In the testing stage, the MFCC feature vectors of the individual music components that were separated by the NMF algorithm were calculated. Figure 9 shows the feature space of 15th separated component

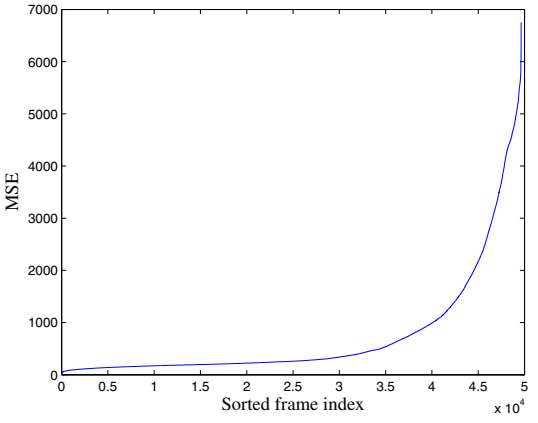

Fig. 11. The MSE values obtained in Figure 10 were sorted from the lower to the higher. The frame indices in the horizontal axis, associated with the MSEs, are shuffled accordingly.

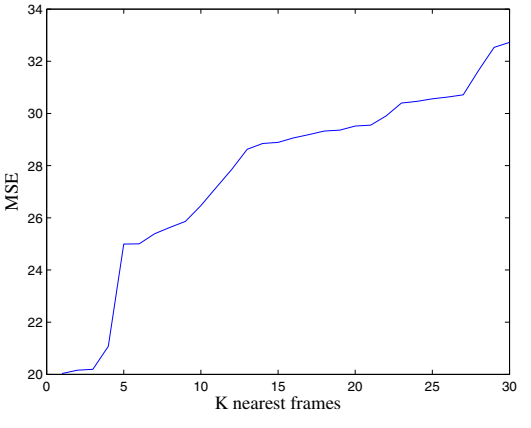

Fig. 12. The MSE values of the K nearest neighbors (i.e. the frames with the K minimal MSEs) are selected based on the K-NN clustering. In this experiment, K was set to 30.

(the final component in our experiment). To determine whether this component belongs to piano or violin, we measured the mean squared error (MSE) between the feature space of the separated component and the feature spaces obtained from the training data. Figure 10 shows the MSEs between the feature vector of a frame (the final frame in this experiment) of the separated component and those obtained in the training data. Then we sort the MSEs according their values along all these frames. The sorted MSEs are shown in Figure 11, where the frame indices were shuffled accordingly. After this, we applied the K-NN algorithm to obtain the 30 neighbors that are nearest to the separated component. The MSEs of these frames are shown in Figure 12. Their corresponding frame indices are shown in Figure 13, from which we can see that all the frame indices are greater

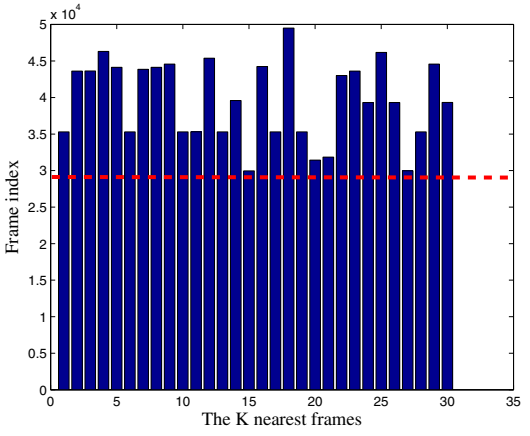

Fig. 13. The frame indices of the 30 nearest neighbors to the frame of the decomposed music note obtained in Figure 12. In our experiment, the maximum frame index for the piano signals is 28971, shown by the dashed line, while the frame indices of violin signals are all greater than 28971. Therefore, this typical audio frame under testing can be classified as a violin signal.

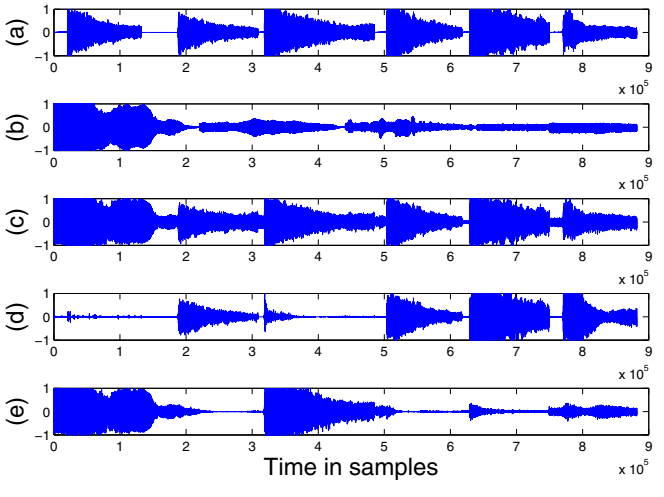

Fig. 14. A separation example of the proposed system. (a) and (b) are the piano and violin sources respectively, (c) is the single channel mixture of these two sources, and (d) and (e) are the separated sources respectively. The vertical axes are the amplitude of the signals.

than 28971, which was the highest index number of the piano signals in the training data. As a result, this component was classified as a violin signal.

Figure 14 shows a separation example of the proposed system, where (a) and (b) are the piano and violin sources respectively, (c) is the single channel mixture

of these two sources, and (d) and (e) are the separated sources respectively. From this figure, we can observe that, although most notes are correctly separated and classified into the corresponding sources, there exist notes that were wrongly classified. The separated notes with the highest SNR is the first note of the violin signal, for which the SNR equals to 9.7dB, while the highest SNR of the note within the piano signal is 6.4dB. The average SNRs for piano and violin are respectively 3.7 dB and 1.3 dB. According to our observation, the separation quality of the notes varies from notes to notes. In average, the separation quality of the piano signal is better than the violin signal.

4 Discussions

At the moment, for the separated components by the NMF algorithm, we calculate their MFCC features in the same way as for the signals in the training data. As a result, the evaluation of the MSEs becomes straightforward, which consequently facilitates the K-NN classification. It is however possible to use the dictionary returned by the NMF algorithm (and possibly the activation coefficients as well) as a set of features. In such a case, the NMF algorithm needs to be applied to the training data in the same way as the separated components obtained in the testing and classification process. Similar to principal component analysis (PCA) which has been widely used to generate features in many classification system, using NMF components directly as features has a great potential. As compared to using the MFCC features, the computational cost associated with the NMF features could be higher due to the iterations required for the NMF algorithms to converge. However, its applicability as a feature for classification deserves further investigation in the future.

Another important issue in applying NMF algorithms is the selection of the mode of the NMF model (i.e. the rank R). In our study, this determines the number of components that will be learned from the signal. In general, for a higher rank R, the NMF algorithm learns the components that are more likely corresponding to individual notes. However, there is a trade-off between the decomposition rank and the computational load, as a larger R incurs a higher computational cost. Also, it is known that NMF produces not only harmonic dictionary components but also sometimes ad-hoc spectral shapes corresponding to drums, transients, residual noise, etc. In our recognition system, these components were treated equally as the harmonic components. In other words, the feature vectors of these components were calculated and evaluated in the same way as the harmonic components. The final decision was made from the labelling scores and the K-NN classification results.

We note that many classification algorithms could also be applied for labelling the separated components, such as the Gaussian Mixture Models (GMMs), which have been used in both automatic speech/speaker recognition and music information retrieval. In this work, we choose the K-NN algorithm due its simplicity. Moreover, the performance of the single channel source separation system developed here is largely dependent on the separated components provided by the

NMF algorithm. Although the music components obtained by the NMF algorithm are somehow sparse, their sparsity is not explicitly controlled. Also, we didn't use the information from the music signals explicitly, such as the pitch information and harmonic structure. According to Li et al. [14], the information of pitch and common amplitude modulation can be used to improve the separation quality. Com

5 Conclusions

We have presented a new system for the single channel music sound separation problem. The system essentially integrates two techniques, automatic note decomposition using NMF, and note classification based on the K-NN algorithm. A main assumption with the proposed system is that we have the prior knowledge about the type of instruments used for producing the music sounds. The simulation results show that the system produces a reasonable performance for this challenging source separation problem. Future works include using more robust classification algorithm to improve the note classification accuracy, and incorporating pitch and common amplitude modulation information into the learning algorithm to improve the separation performance of the proposed system.

References

1. Abdallah, S.A., Plumbley, M.D.: Polyphonic Transcription by Non-Negative Sparse Coding of Power Spectra. In: International Conference on Music Information Retrieval, Barcelona, Spain (October 2004)
2. Barry, D., Lawlor, B., Coyle, E.: Real-time Sound Source Separation: Azimuth Discrimination and Re-synthesis, AES (2004)
3. Brown, G.J., Cooke, M.P.: Perceptual Grouping of Musical Sounds: A Computational Model. J. New Music Res. 23, 107–132 (1994)
4. Casey, M.A., Westner, W.: Separation of Mixed Audio Sources by Independent Subspace Analysis. In: Proc. Int. Comput. Music Conf. (2000)
5. Devijver, P.A., Kittler, J.: Pattern Recognition - A Statistical Approach. Prentice Hall International, Englewood Cliffs (1982)
6. Every, M.R., Szymanski, J.E.: Separation of Synchronous Pitched Notes by Spectral Filtering of Harmonics. IEEE Trans. Audio Speech Lang. Process. 14, 1845–1856 (2006)
7. Fevotte, C., Bertin, N., Durrieu, J.-L.: Nonnegative Matrix Factorization With the Itakura-Saito Divergence. With Application to Music Analysis. Neural Computation 21, 793–830 (2009)
8. FitzGerald, D., Cranitch, M., Coyle, E.: Extended Nonnegative Tensor Factorisation Models for Musical Sound Source Separation, Article ID 872425, 15 pages (2008)
9. Fukunage, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press Inc., London (1990)

10. Gutierrez-Osuna, R.: Lecture 12: K Nearest Neighbor Classifier, http://research.cs.tamu.edu/prism/lectures (accessed January 17, 2010)
11. Hoyer, P.: Non-Negative Sparse Coding. In: IEEE Workshop on Networks for Signal Processing XII, Martigny, Switzerland (2002)
12. Lee, D.D., Seung, H.S.: Learning the Parts of Objects by Non-Negative Matrix Factorization. Nature 401, 788–791 (1999)
13. Lee, D.D., Seung, H.S.: Algorithms for Non-negative Matrix Factorization. In: Neural Information Processing Systems, Denver (2001)
14. Li, Y., Woodruff, J., Wang, D.L.: Monaural Musical Sound Separation Based on Pitch and Common Amplitude Modulation. IEEE Transactions on Audio, Speech, and Language Processing 17, 1361–1371 (2009)
15. Mellinger, D.K.: Event Formation and Separation in Musical Sound. PhD dissertation, Dept. of Comput. Sci., Standford Univ., Standford, CA (1991)
16. Opolko, F., Wapnick, J.: McGill University master samples, McGill Univ., Montreal, QC, Canada, Tech. Rep. (1987)
17. Pedersen, M.S., Wang, D.L., Larsen, J., Kjems, U.: Two-Microphone Separation of Speech Mixtures. IEEE Trans. on Neural Networks 19, 475–492 (2008)
18. Rickard, S., Balan, R., Rosca, J.: Real-time Time-Frequency based Blind Source Separation. In: 3rd International Conference on Independent Component Analysis and Blind Source Separation, San Diego, CA (December 2001)
19. Smaragdis, P., Brown, J.C.: Non-negative Matrix Factorization for Polyphonic Music Transcription. In: Proc. IEEE Int. Workshop Application on Signal Process. Audio Acoust., pp. 177–180 (2003)
20. Smaragdis, P.: Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs. In: Puntonet, C.G., Prieto, A.G. (eds.) ICA 2004. LNCS, vol. 3195, pp. 494–499. Springer, Heidelberg (2004)
21. The University of Iowa Musical Instrument Samples Database, http://theremin.music.uiowa.edu
22. Virtanen, T.: Sound Source Separation Using Sparse Coding with Temporal Continuity Objective. In: International Computer Music Conference, Singapore (2003)
23. Virtanen, T.: Separation of Sound Sources by Convolutive Sparse Coding. In: Proceedings of ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing, Jeju, Korea (2004)
24. Virtanen, T.: Sound Source Separation in Monaural Music Signals. PhD dissertation, Tampere Univ. of Technol., Tampere, Finland (2006)
25. Virtanen, T.: Monaural Sound Source Separation by Non-Negative Matrix Factorization with Temporal Continuity and Sparseness Criteria. IEEE Transactions on Audio, Speech, and Language Processing 15, 1066–1073 (2007)
26. Wang, D.L., Brown, G.J.: Computational Auditory Scene Analysis: Principles, Algorithms, and Applications. Wiley/IEEE Press (2006)
27. Wang, B., Plumbley, M.D.: Investigating Single-Channel Audio Source Separation Methods based on Non-negative Matrix Factorization. In: Nandi, Zhu (eds.) Proceedings of the ICA Research Network International Workshop, pp. 17–20 (2006)
28. Wang, B., Plumbley, M.D.: Single Channel Audio Separation by Non-negative Matrix Factorization. In: Digital Music Research Network One-day Workshop (DMRN+1), London (2006)

29. Wang, W., Luo, Y., Chambers, J.A., Sanei, S.: Note Onset Detection via Non-negative Factorization of Magnitude Spectrum. EURASIP Journal on Advances in Signal Processing, Article ID 231367, 15 pages (June 2008); doi:10.1155/2008/231367
30. Wang, W., Cichocki, A., Chambers, J.A.: A Multiplicative Algorithm for Convolutive Non-negative Matrix Factorization Based on Squared Euclidean Distance. IEEE Transactions on Signal Processing 57, 2858–2864 (2009)
31. Webb, A.: Statistical Pattern Recognition, 2nd edn. Wiley, New York (2005)
32. Woodruff, J., Pardo, B.: Using Pitch, Amplitude Modulation and Spatial Cues for Separation of Harmonic Instruments from Stereo Music Recordings. EURASIP J. Adv. Signal Process. (2007)