2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Pitch Estimation Using Non-negative Matrix Factorization

Ryan Burt, Goktug T. Cinar, José C. Principe
Computational NeuroEngineering Laboratory
Department of Electrical and Computer Engineering
University of Florida
Gainesville, Florida 32601

***Abstract*—The problem of pitch detection consists of estimating the dominant frequency present in a certain time window. This paper demonstrates and analyzes the use of a non-negative matrix factorization technique with a frequency basis formed with a correntropy kernel. This offers the advantage that the frequency basis is adaptable, allowing the matrix factorization to fit the data precisely, as well as including a dictionary specifically to account for noise. Using non-negative matrix factorization also allows an increase in dimensionality, which increases the frequency resolution of the algorithm.**

The method is tested on a database of trumpet notes and compared to other current methods, improving on their performance for noisy signals.

***Index Terms* - Correntropy, non-negative matrix factorization, pitch detection, spectral representation**

I. Introduction

A fundamental part of music transcription is pitch detection for note determination. This can be accomplished through various methods, including both time domain and frequency domain analysis. One common time domain technique is the autocorrelation function, which is a measure of similarity. The lags at which peaks occur in the autocorrelation function correspond to the dominant frequencies found in the signal. The maximum peak in the autocorrelation function (with a nonzero lag) can be used to find the dominant frequency in the signal, which is then used to classify the note.

$$R[m] = \frac{1}{N-m+1}\sum_{n=m}^{N} x[n]x[n+m] \tag{1}$$

The autocorrelation function only takes second order moments into account, so one way to improve upon it is by using a function that also uses higher order moments. This is accomplished using a correntropy kernel, which adds a free parameter (the kernel size) that determines the combination of moments in the resulting signal. The correntropy function has the advantage of being much peakier than autocorrelation, which helps avoid misclassification due to overlapping peaks [3],[6].

$$V[m] = \frac{1}{N-m+1}\sum_{n=m}^{N} \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{||x-y||^2}{2\sigma^2}) \tag{2}$$

Another set of techniques used to estimate the dominant frequencies in a signal is spectral estimation. These take the signal and estimate the power spectral density of the signal, from which the dominant frequencies can be determined. The power spectral density is found by taking or approximating the Fourier transform of the autocorrelation of the signal.

Although the autocorrelation function has been used successfully with spectral estimation techniques, the autocorrentropy function poses an interesting problem. The inclusion of extra moments in the new signal creates more peaks in the frequency domain [11]. Where the spectrum of the autocorrelation function of a pure sinusoid will only have a peak at that specific frequency, the spectrum of the autocorrentropy function will have many more resulting peaks at harmonics of that frequency with their respective sizes depending on the kernel and parameters used.

This paper studies a proposed method for estimating the power spectral density using the autocorrentropy function, called the correntropy spectral density. Using non-negative matrix factorization with a correntropy based frequency dictionary, a representation of the correntropy spectral density is obtained that avoids the problem of adding harmonics in the frequency domain. This algorithm is then tested as a pitch detection algorithm on a database of musical notes.

In this paper, the use of a new correntropy spectral estimator is studied. Section II presents the method used to estimate the spectrum. Section III contains results comparing the new method with established pitch detection algorithms. Finally, Section IV concludes the paper and lays the basis for future work on this topic.

II. Method

This section details the method used to create a new spectral estimator using the autocorrentropy function of a signal instead of the autocorrelation. Whereas previous autocorrentropy spectrum suffered from multiple harmonics, this new technique eliminates those while providing the benefits of using the correntropy function, namely peakiness and data from higher order moments.

A. Non-Negative Matrix Factorization

The principle behind non-negative matrix factorization is that a matrix can be approximated as the product of two matrices whose outer dimensions are the same as the original matrix. That is,

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

$$V \approx \hat{V} = WH \tag{3}$$

where W is defined as the dictionary matrix, and H as the coefficient matrix [7]. If V has dimensions M x P, then we define W as having dimension M x K and H as having K x P, so the product between W and H has the same dimensions as the original matrix. The ability to set K is powerful: setting K low reduces dimensionality and improves processing time, but setting K high (especially in the case where W is a frequency dictionary) results in an overcomplete dictionary, which leads to sparser results. For this purpose, V is the autocorrentropy function of the original signal instead of the signal itself.

$$f_k = \frac{k}{TL}, k = 0, ..., \frac{LN}{2} \tag{4}$$

Eq. (4) shows the frequency grid, with T being the time span of the input signal and L as the overcompleteness parameter [11]. As L is increased over one, the frequencies as sampled more finely than in a standard frequency transform, resulting in a high resolution spectrum.

In standard non-negative matrix factorization, a cost function is defined and then the matrices are updated in order to minimize this cost function.

$$\min L(V||WH) \tag{5}$$

W and H are both defined to be strictly non-negative. The cost function used in this case is the Frobenius norm, which is defined as:

$$D_F(V||WH) = \frac{1}{2}\sum_{ij}(V_{ij} - [WH]_{ij})^2 \tag{6}$$

Since both W and H are optimized to approximate V, this is normally split into two separate problems, where W and H are optimized separately in alternating steps using gradient descent [8].

$$\min \frac{1}{2}||H^T W^T - V^T||_F^2 \tag{7}$$

$$\min \frac{1}{2}||V - WH||_F^2 \tag{8}$$

As an alternative to traditional gradient descent algorithms, two multiplicative update steps can be used. These alternative steps actually function as gradient descent algorithms with an adaptive step size [9].

$$H_{ij}^{new} = H_{ij}^{old}\frac{(W^T V)_{ij}}{(W^T WH)_{ij}} \tag{9}$$

$$W_{ij}^{new} = W_{ij}^{old}\frac{(VH^T)_{ij}}{(WH^T H)_{ij}} \tag{10}$$

This multiplicative update actually serves as a gradient descent update with an adaptive step size based on the current state of the matrices. These update steps are then performed until the multiplication of W and H approximate V within specifications.

B. Noise Dictionary

The noise dictionary is the first addition to standard spectral estimation using non-negative matrix factorization. The noise dictionary consists of an identity matrix the same size as the target vector; this serves as a kronecker delta dictionary to account for any noise in the signal. This allows the frequency dictionary to adapt to the signal while the noise dictionary adapts to any anomalies that the frequency dictionary cannot account for, leading to much lower fitting errors [10] [11].

With the addition of the noise dictionary, Eq. (3) now becomes,

$$V \approx \hat{V} = [W_1, W_2][H_1, H_2]^T \tag{11}$$

where H_1 is the coefficient matrix associated with the frequency dictionary and H_2 is the coefficient matrix associated with the noise.

C. Frequency Dictionary Update Procedure

For the specific case of frequency estimation, the dictionary matrix W is initialized with frequency atoms, usually based on the sine and cosine functions [10]. When the dictionary matrix is initialized in this manner, it cannot be updated by a gradient descent method, because doing so would destroy the frequency basis of the matrix. Instead, only the coefficient matrix H is updated to minimize the cost function.

For this implementation, the dictionary matrix was changed to contain frequency atoms based on a generalized correntropy kernel. In the majority of frequency based work for non-negative matrix factorization, only the coefficient matrix could be updated due to the definition of the frequency matrix [6] due to the update destroying the frequency of each atom. However, by defining the frequency matrix using a variant of the Gaussian kernel, a pure frequency dictionary is created where every atom has a free parameter: the kernel size [1].

$$W_1(n,k) = \exp(-\frac{2\sin^2(\pi f_k \tau_n)}{\sigma_k^2}) \tag{12}$$

In this kernel, f_k is the frequency of the atom and τ_n is the nth lag of the function. By initializing the frequency dictionary with this kernel (using the initial kernel size given by Silvermans Rule), it is possible to now update the frequency dictionary using a gradient descent algorithm by changing only the kernel size. This will not change the frequency content of the dictionary matrix, but will adapt the shape of the atoms to better fit the data. This allows the fit to be sparser and more localized. The following three equations show the update procedure for the kernel sizes [11].

$$\nabla_k = \frac{4H_1(k)}{\sigma_k^2}\sum_{m=1}^{M}\epsilon_m W_1(m,k)\sin^2(\pi\tau_m f_k) \tag{13}$$

$$\epsilon_m = \sum_{i=1}^{K} H_1(i)W_1(m,i) + \sum_{j=1}^{M} H_2(j)W_2(m,j) - v_m \tag{14}$$

$$\sigma_k^{t+1} - \sigma_k^t = \Delta\sigma_k^{t+1} = \alpha\Delta\sigma_k^t - \mu\nabla_k \tag{15}$$

Here, v_m is the initial autocorrentropy value at lag m, and α and μ are the momentum and learning parameters respectively. This frequency dictionary update is important because it automatically selects the kernel size used for each frequency in the dictionary. Instead of searching across a range of kernels, the dictionary uses gradient descent to find the proper kernel size at each frequency to match the initial autocorrentropy function, which is estimated using the value given by Silverman's Rule.

D. Algorithm

To determine the pitch of a musical note, a step by step process can be followed to use correntropy spectral estimation; namely, the correntropy-based non-negative matrix factorization (CNMF). First, the autocorrentropy function of the time series is estimated using Eq. (2). This functions as the V that the non-negative matrix factorization will approximate. Once this has been found, W_1 and W_2 are initialized to Equation (8) and as the identity matrix of size MxM, respectively. H is the initialized according to Eq. (14) and then split into H_1 and H_2 based on their respective sizes.

$$H^0 = [W_1^0 W_2^0]^T V \tag{16}$$

Once the dictionary and coefficient matrices have been initialized, non-negative matrix factorization is performed, alternating the multiplicative update in Eq. (8) for the total coefficient matrix H and the kernel size update outlined in the previous subsection for the dictionary matrix W_1. Since W_2 is a kronecker delta dictionary meant to handle noise, it requires no update. The dominant frequency in the signal can be found once the update finishes by simply searching for the maximum value in H_1 and finding the frequency in the corresponding frequency atom in the dictionary matrix.

This algorithm adds three parameters that need to be chosen by the user: L, the overcompleteness parameter, μ, the learning rate, and α, the momentum rate. The momentum and learning rates affect only the convergence of the algorithm. The overcompleteness parameter, if set higher than 1, should increase performance by sampling the frequency domain finer, leader to more precise results.

III. Results

The algorithm was tested on the University of Iowa's musical instrument samples, specifically on the non-vibrato trumpet sampled at medium volume. The notes were down-sampled from the standard 44100Hz to 11025Hz. Each note was then broken into windows consisting of 800 samples and standardized before testing by subtracting the mean and dividing by the standard deviation of each window.

Figure (1) shows the power spectral density and correntropy spectral density of an E5 note played by a trumpet (with and without noise). Though they both show the same dominant frequency in the signal, the CSD has extremely narrow peaks due to the overcompleteness of the algorithm, leading to the fine frequency resolution. Also, the magnitude of all non-dominant peaks in the CSD are relatively small compared to the secondary peaks in the PSD, which is a result of using correntropy. Both of these results could prove useful in a pitch detection algorithm - the narrower peaks prevent interference from two close notes, while smaller harmonics ensure that the dominant frequency is not overwhelmed by the addition of secondary peaks.

SNR (dB)	20	5	1
SWIPE	.8457	.0536	.0328
Yin	.9781	.6947	.3611
CNMFS No W_2	.7352	.1663	.0427
CNMFS	.9519	.8173	.5875

TABLE I
RESULTS WITH ZERO MEAN WHITE NOISE

SNR (dB)	20	5	1
SWIPE	.7681	.0317	.0317
Yin	.9781	.4628	.2177
CNMFS No W_2	.7133	.0613	.0295
CNMFS	.9617	.7429	.3972

TABLE II
RESULTS WITH ZERO MEAN OUTLIER NOISE

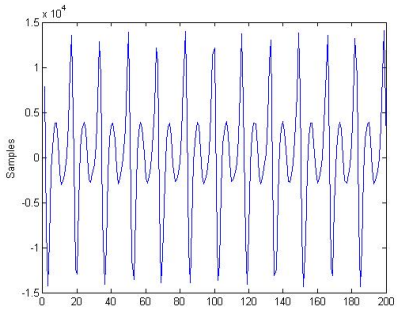
One interesting note is that the dominant peak in the PSD spectrums is the second harmonic of E5 instead of E5. This is a clear example of the advantages of using CSD, which uses the kernel to help eliminate the harmonics present in the spectrum. A drawback of the CSD, however, is that the spectrum does contain more small peaks and fluctuations, which could be due to the approximation inherent in non-negative matrix factorization.

Also, the outlier noise has a much greater visual effect on the PSD as opposed to the CSD. With non-zero mean outlier noise with SNR = 1 added to the signal, the CSD shows a few slightly increased secondary peaks and some fluctuation in the low frequencies. The PSD, however, shows increased power at every frequency in the spectrum and numerous small peaks added.

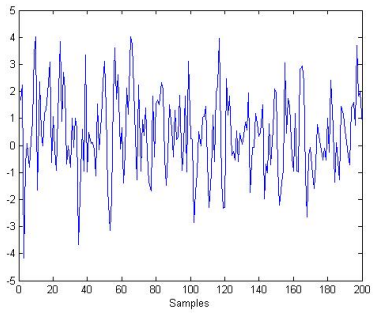
In addition to the CNMF algorithm, the data was tested with two state-of-the-art methods for comparison: SWIPE and Yin. These techniques have both been proven to produce exceptional results on monophonic music data, with steady state note detections above 95% for musical notes [5][6].

After conducting the basic tests, noise was added in three forms at different levels. The noise types were white noise with a Gaussian distribution, the same white noise with 20% outliers from a second distribution still with zero mean, and white noise with 20% outliers and a non-zero mean. The parameters used for CNMF are $L = 10$, $\mu = .05$, and $\alpha = .05$. The algorithm was tested in two versions: one with the noise dictionary (W_2) and one without to study the affect the dictionary has on adaptation to noisy signals.

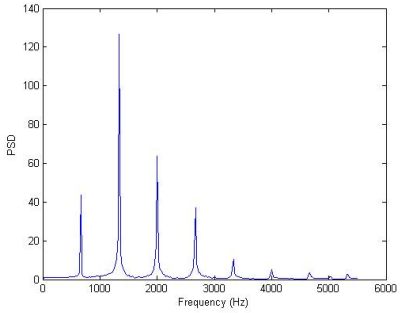
These results show that the CNMF algorithm with the noise dictionary performs consistently better than both Yin and SWIPE once noise is introduced to the system, and is at least on par with both when the noise level is low. When the



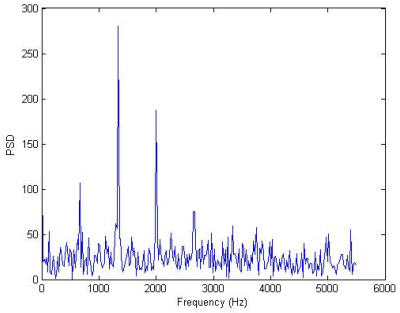
(a) The clean trumpet note



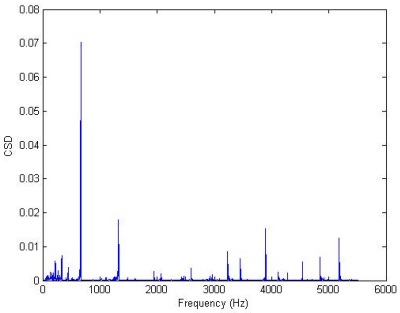
(b) The trumpet note with non-zero outlier noise (SNR = 1)



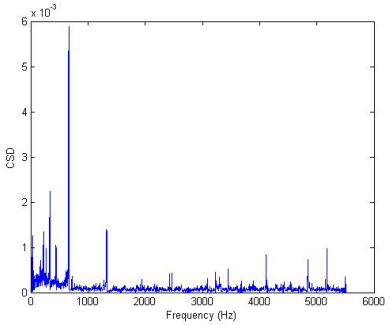
(c) PSD of the clean note



(d) PSD of the noisy note



(e) CSD of the clean note



(f) CSD of the noisy note

Fig. 1. PSD and CSD results on an E5 note played by a trumpet.

SNR (dB)	20	5	1
SWIPE	.7867	.0317	.0219
Yin	.9781	.4737	.2451
CNMFS No W_2	.7330	.0744	.0416
CNMFS	.9606	.7418	.4136

TABLE III
RESULTS WITH NON-ZERO MEAN OUTLIER NOISE

noise dictionary is omitted from the CNMF algorithm, it loses much of its advantage over the other algorithms, performing similarly to SWIPE.

IV. CONCLUSION

This paper proposes the use of a correntropy based non-negative matrix factorization technique for music note transcription. By using correntropy kernels for the frequency basis, the dictionary matrix can easily be adapted to fit the data without altering the frequency structure of the matrix. In addition, the use of a second dictionary to account for noise makes this method extremely insensitive to even high noise levels.

This technique provides an overcomplete spectral representation of the correntropy function, with the benefit of eliminating the harmonics inherent in estimating the Fourier Transform of the correntropy function. Used as a pitch detection algorithm, the results compare favorably with other current techniques, such as SWIPE and Yin.

For future work, the parameter space will be studied so the effect of choosing different parameters on the CSD is known. Specifically, this will involve choosing different overcompleteness parameters, window sizes, stopping criteria, and momentum and learning parameters. Once more is known about how these parameters affect the CSD, CNMFS will

become a much more powerful tool in pitch detection and possibly other applications.

A future application for this algorithm includes testing it on polyphonic sound sources. CNMF should lend itself well to polyphonic music transcription due to the peakiness of the data and lack of harmonics leading to less overlap, as well as the ability to finely sample the frequencies. In addition to simple note frequency estimation in polyphonic sources, this algorithm may lend itself to note onset/offset detection by simply finding the time windows where frequencies are above a certain power threshold.

V. Acknowledgments

This work is supported by the Office of Naval Research (ONR) grant #N000141010375.

References

[1] P. Huijse, P. A. Estevez, P. Protopapas, P. Zegers, and J. C. Principe, "An Information Theoretic Algorithm for Finding Periodicities in Stellar Light Curves" *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5135–5145, 2012.

[2] E. Vincent, N. Bertin, and R. Badeau, "Adaptive Harmonic Spectral Decomposition for Multiple Pitch Estimation" *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 528–537, Mar. 2010.

[3] J. Xu and J. Principe, "A Novel Pitch Determination Algorithm Based on Generalized Correlation Function" *Machine Learning for Signal Processing*, Mar. 2007.

[4] A. de Cheveigne and H. Kawahara, "YIN, a Fundamental Frequency Estimator for Speech and Music" *Journal of Acoustics*, pp. 1917–1930, 2002.

[5] A. Camacho, "SWIPE: A Sawtooth Waveform Inspired Estimator for Speech and Music" Ph.D. Disseration, Univeristy of Florida, 2007

[6] J. Principe, "Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives " Spring Verlag, 2010

[7] Y. Wang and Y. Zhang, "Nonnegative Matrix Factorization: A comprehensive Review" *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 6, pp. 1336–1353, 2013.

[8] C. Lin, "Projected Gradient Methods for Nonnegative Matrix Factorization" *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

[9] Z. Yang and E. Oja, "Unified Development of Multiplicative Algorithms for Linear and Quadratic Nonnegative Matrix Factorization" *Transactions on Neural Networks*, vol. 22, pp. 1878–1891, Dec 2011.

[10] S. S. Chen and D. Donoho, "Application of Basis Pursuit in Spectrum Estimation" *Acoustics, Speech, and Signal Processing, 1998. Proceedings of the 1998 IEEE Conference on*, vol. 3, pp. 1865–1868, 1998.

[11] P. Huijse, P. Estevez, J. Principe, P. Protopapas, and P. Zegers, "Non-Negative Matrix Factorization for Correntropy Spectral Density" *IEEE Transactions on Signal Processing*, under review, 2013