Energy distribution for Coefficients of Redundant Signal Representations of Music

Line Ørtoft Endelt and Anders la Cour-Harbo
Aalborg University
Department of Control Engineering
Frb. Vej 7C, 9220 Aalborg East, Denmark
{oertoft, alc}@control.aau.dk

Abstract

In this paper we investigate how the energy is distributed in the coefficients vector of various redundant signal representations of music signals. The representations are found using Basis Pursuit, Matching Pursuit, Alternating Projections, Best Orthogonal Basis and Method of Frames, with five different time-frequency dictionaries. We have applied these methods to music to examine their ability to express music signals in a sparse manner for a number of dictionaries and window lengths. The evaluation is based on the m-term approximation needed to represent 90 %, 95 %, 99 % and 99.9 % of the energy in the coefficients, also the time consummation for finding the representations are considered. The distribution of energy in the coefficients of the representations found using Basis Pursuit, Matching Pursuit, Alternating Projections and Best Orthogonal Basis depends mainly on the signal, and less on the minimization method, the dictionary and the length of the analysis window.

The results indicate, that the sparseness of the representations do indeed tell something about the music signal, and this is an interesting subject for further investigation.

1 Introduction

The results presented here are obtained as part of a research project on Automatic Classification of Music. We are interested in finding sparse representations of music signals, based on the idea that we can find good features from a sparse representation of a music signals, that capture "the nature" or significant elements of each particular piece of music. This approach is in some sense opposite compared to other approaches on audio feature extraction (e.g. [5], [8], [9] and [6]), in that we instead of trying to detect particular elements in the signal investigate what a variety of different redundant signal representation can reveal about the signal.

When applying e.g. a Fourier or Wavelet Transform to describe a signal, a complete description is achieve which is (usually) much more sparse, than the original signal, and more meaningful (intuitive) in terms of what the signals contains. But even though the description is complete, it is not necessarily good and/or sparse. A music signal often have both "pure" frequencies, which are described best by harmonic functions, and contains elements, e.g. a drumbeat, which at the onset time is best described by wavelets. Therefore we are investigating what can be achieved in terms of feature extraction by representing music signals in a redundant set of functions.

The focus of this paper is on the sparseness achieved when combining different optimization methods with different sets of functions (dictionaries) and signal lengths.

2 Methodology

The music signals are considered to be elements in $\boldsymbol{R}^N$. A representation of a signal $\boldsymbol{b} \in \boldsymbol{R}^N$ in a dictionary of size k, is a vector $\boldsymbol{x} \in \boldsymbol{R}^k$, satisfying

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{1}$$

where $\boldsymbol{A}$ is an $n \times k$ matrix having the atoms in the dictionary as its columns. The vector $\boldsymbol{x}$ contains the k coefficients of the representation. When the dictionary contains more than n elements, this representation is in general not unique. This on one hand leads to added flexibility in choice of representation, on the other leads to higher complexity in finding the possible representations.

Five different minimization methods for finding representations as in (1) are applied; Basis Pursuit, BP [1], which minimizes the ℓ^1 norm of the coefficient vector. Matching Pursuit, MP [7], which successively picks the

	average of $\bar{f}_{90}$					average of $\bar{f}_{99.9}$				
	BP	AP	MP	BOB	MOF	BP	AP	MP	BOB	MOF
DCT	0.0178		0.0158		0.0394	0.3097		0.2395		0.5734
DCT, WP	0.0153	0.0192	0.0161		0.5851	0.3244	0.2999	0.2201		3.1065
WP	0.0320	0.0350	0.0304	0.0350	0.5906	0.4033	0.3003	0.2552	0.3011	2.8374
CP	0.0167	0.0184	0.0138	0.0186	0.1620	0.3162	0.2691	0.2033	0.2732	1.8940
WP,CP	0.0155	0.0343	0.0133		0.6647	0.3065	0.3163	0.1849		4.7200

Table 1. The average, over all songs and all window lengths, of $\bar{f}_{90}$ and $\bar{f}_{99.9}$, for the combinations of dictionary and minimization methods.

atoms with the largest correlation with the signal subtracted the projection on to the former chosen atoms. Alternating Projections, AP, where the signal is decomposed alternating between the bases of the dictionary each time picking the most significant atoms. Best Orthogonal Basis, BOB [10], chooses the best basis among all the bases, and in the present research best means having the smallest ℓ^1 norm (this method is applied only to CP and WP Dictionaries) and Method of Frames MOF [2], which minimizes the ℓ^2 norm of the coefficient vector.

The dictionaries applied are constructed from the following three transforms. Discrete Cosine Transform (DCT), a Wavelet Packet (WP) and a Cosine Packet (CP). The WP is generated using the coiflet wavelet with filter length 12. The choice of wavelet is based on the results in [3], even though the results there are obtained using a DWT, it is reasonable to assume that not much is gained in sparseness by increasing the smoothness of the wavelet beyond the smoothness of the signal, when applied in a WP. The CP contains locally trigonometric cosine functions generated with a "sine bell".

The dictionaries applied can be seen in the first column of Table 1. The DCT is over samples by a factor two in the first dictionary and the WP and CP are applied with redundancy $\log_2(N)$-5, corresponding to the first $\log_2(N)$-5 levels. Note that the dictionaries contain many orthonormal bases, which are essential for some of the minimization methods.

These three sub-dictionaries describe different elements in a music signal. The DCT describes frequencies over the whole time interval, while the CP describes frequencies over local dyadic intervals of the signal. The WP is good at describing both rapid changes (short duration events) in the signal, which appears at e.g. a note onset, and long duration events, without changing representation [4].

The representation methods are applied on 30 sec. sequences from 136 pieces of music, from a number of different genres. The sample frequency is 44.1 kHz, and the sampling is started 60 sec. after the beginning of the song. The 30 sec. music sequences are divided into windows of length 256, 1024, 4096 and 8192 samples (see Figure 1), corresponding to 6 ms, 23 ms, 93 ms and 186 ms. This results in respectively 5167, 1291, 322 and 161 analysis windows.

Figure 1. The 30 sec. music sequences are divided into successive but non overlapping windows of length 256, 1024, 4096 or 8192 samples.

All the combinations of minimization methods and dictionaries (except four combinations, which can be seen implicit in Table 1) are applied with the four different window lengths on all 136 pieces of music. Which gives 72 different calculation combinations.

The representations are found using already existing Matlab functions (for references see [1] and http://www.control.aau.dk/∼alc/Homemade/apro.m), which have been adjusted to this test setup. The calculations are performed as distributed computations on a number of PCs. All calculations for one piece of music is performed on the same PC, so it is possible to compare the computation times.

Storing all the coefficients for all the representations requires disc space in the order of tera bytes. Consequently, we have computed the m-term approximation needed to represent 90 %, 95 %, 99 % and 99.9 % of the energy in the coefficients. It is denoted m_β, where β corresponds, to the fraction of energy. The m-term approximation, will be represented relative to the window length as

$$f_\beta \quad = \quad \frac{m_\beta}{N},$$

where N is the signal length. The average over all the windows in the 30 sec. music pieces, will be denoted $\bar{f}_\beta$.

3 Results

In Figure 1 the measures $\bar{f}_{90}$, $\bar{f}_{95}$, $\bar{f}_{99}$ and $\bar{f}_{99.9}$ are shown for all combinations of window length, dictionary

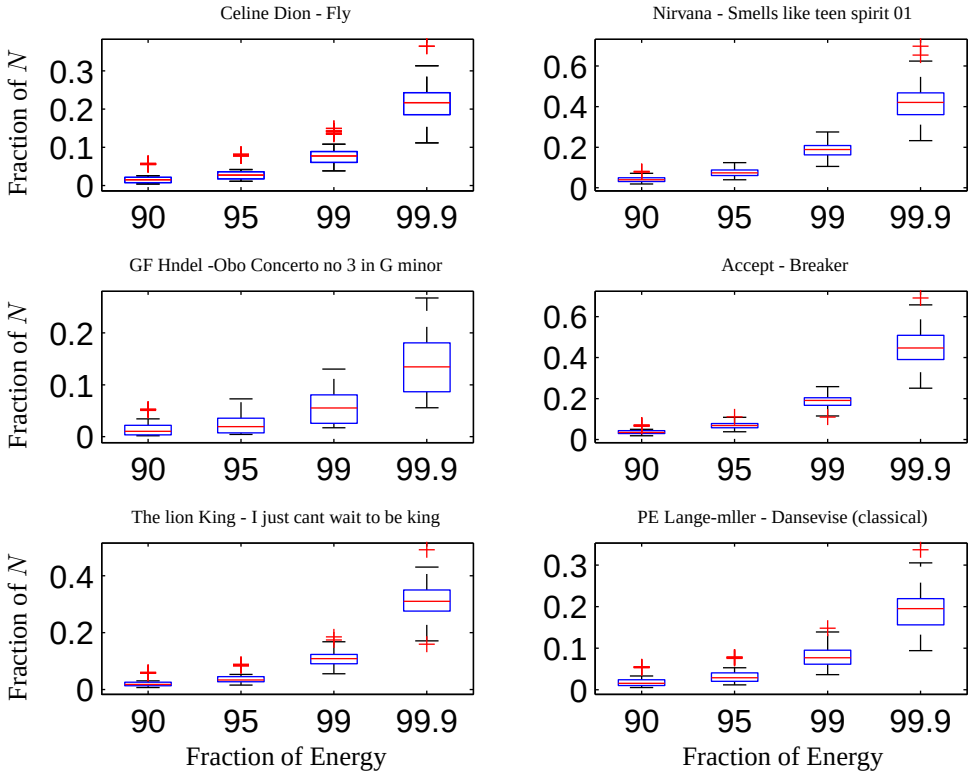

Figure 2. The measure $\bar{f}_\beta$ versus the fraction of energy in the coefficient vector. Each plot corresponds to one song, and results are included for all combinations of window length, dictionary and minimization method, except MOF. The box itself extends from the lower quartile value through the median (middle notch) to the upper quartile value. The whiskers show the range of the remaining values, except for outliers which are more than 1.5 $\times$ inter-quartile range away from lower or upper quartile values. Outliers are marked with +. Note, that the scale on the x-axis is not linear, and the scale on the y-axis is different for the six plots.

and minimization method except MOF (since the $\bar{f}_\beta$ values are significantly larger). For $\bar{f}_{90}$, $\bar{f}_{95}$ and $\bar{f}_{99}$ the values lie fairly close for a particular piece of music. The fraction of coefficients needed to describe 99.9 % of the energy vary over a larger range, the median vary between 15 % and 45 % for the six results shown.

To compare the methods an average of $\bar{f}_{90}$ and $\bar{f}_{99.9}$ for all combinations of minimization methods and dictionaries has been calculated over all 136 songs and all window lengths (see Table 1). Even though the value of the measurements vary much for the different songs, this do give meaning, since the distribution of the measures between the dictionaries and the minimization methods are very much alike for the different pieces of music.

Figure 3 shows the average computation times (over the 136 different music sequences) for all 72 combinations of minimization method, dictionary and signal length. The computation times for the methods BOB and MOF mainly depends on the dictionary which is in agreement with the complexity N (see [1]), since the number of windows is inverse proportional to the window length. The computation times for the methods MP and BP agree with a complexity of $N \log N$ (see [1]). For AP the computation time decreases with the signal length, it has not been investigated, but it might be because relatively fewer analysis operations are needed for longer signals.

The average of the computation times over all music sequences and windows lengths, for finding the representations of a 30 sec. music sequences for all combinations of dictionaries and minimization methods are shown in Table 2. Even though taking the average over the signal lengths for MP and BP do not give much meaning, this values still gives ground for comparing the computation time of the different minimization methods.

The two dictionaries WP and CP are applied with the same redundancy, so the large difference in the computa-

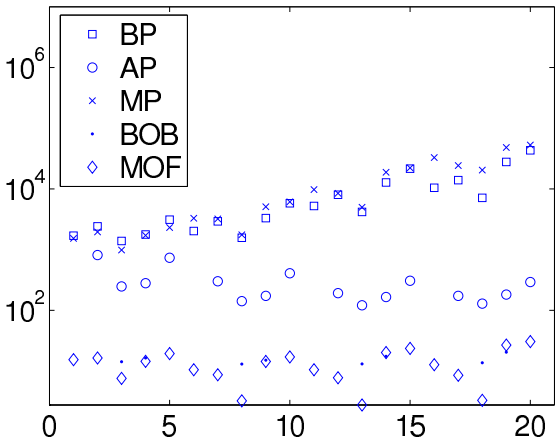

Figure 3. The average computation time for a 30 sec. music sequence. The y axis is time in seconds, and the x-axis is the 20 combinations of time and dictionary. The first five corresponds to the five dictionaries for signal length 256, the next five is the five dictionaries combined with length 1024 and so on.

	BP	AP	MP	BOB	MOF
DCT	1.3498		3.3094		0.0034
DCT, WP	1.9022	0.1030	2.6425		0.0029
WP	0.9940	0.0445	1.9610	0.0038	0.0012
CP	3.1783	0.0557	5.1358	0.0048	0.0053
WP, CP	5.1298	0.1213	5.8201		0.0063

Table 2. The average over the 136 different songs, and the four different window lengths of the computation time (in hours), for finding the representations of a 30 sec. music sequences for the combinations of dictionaries and minimization methods.

tion times for most minimization methods must be due to the implementations of the dictionaries.

4 Discussion

It can be seen that MP in general performs better than the other methods, meaning that the energy is contained in a smaller fraction of the coefficients. The representations found using MP do not satisfy equation (1), but only describe (at least) 99 % of the energy in the original signal, hence all "the small" coefficients required to have a complete description of the original signal is left out, which might give this seemingly better sparseness.

MOF has a tendency to spread out the energy, hence it often requires more coefficients than the signal length, and it gets worse, when the redundancy in the dictionary is increased.

In general the Wavelet Packet performs worse, when it is not combined with a frequency dictionary, this is due to the "high concentration" of relative pure frequencies in music. An advantage is that the computation time is smaller than for the other dictionaries.

The main difference in the values of $\bar{f}_\beta$ lies in the signal, as can be seen in Figure 1. It can be seen that for the "Nirvana" and "Accept" pieces, the values are in general high, and for the others which are more mellow, the values are lower. This indicate, that the concentration of the coefficients (or sparseness of the representation) do depend on which type of music is considered.

The calculation times are of cause too large to use for any realistic music classification system, but if only the information for a few seconds (or ms) is needed, which other audio classification systems indicate ([5],[9] and [6]), it is realistic to used these methods when "real time" is not required.

5 Acknowledgment

This work is supported by the Danish Technical Science Foundation, program no. 56-00-0143.

References

[1] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Comput.*, 20(1):33–61, 1998.

[2] I. Daubechies. Time-frequency localization operators: A geometric phase space approach. *IEEE Trans. Inform. Theory*, pages 605–612, 1988.

[3] L. Ø. Endelt and A. la Cour-Harbo. Wavelets for sparse representation of music. In *Proceedings of Wedelmusic2004*, 2004.

[4] A. Jensen and A. Cour-Harbo. *Ripples in Mathematics: The discrete wavelet transform*. Springer, 2001.

[5] S. Z. Li. Content-based audio classification and retrieval using the nearest feature line method. *IEEE Transactions on Speech and Audio Processing*, 8(5):619–625, September 2000.

[6] C.-C. Lin, S.-H. Chen, T.-K. Truong, and Y. Chang. Audio classification and categorization based on wavelets and support vector machine. *IEEE Transactions on Speech and Audio Processing*, 13(5), September 2005.

[7] S. Mallet and Z. Zhang. Matching pursuit in a time-frequency dictionary. *IEEE Transactions on Signal Processing*, pages 3397–3415, 1993.

[8] E. Scheirer. Tempo and beat analysis of acoustic musical signals. *Journal of Acoustical Society of America*, pages 419–429, January 1998.

[9] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, July 2002.

[10] M. V. Wickerhauser. *Adapted Wavelet Analysis from Theory to Software*. A K Peters, 1994.