AGGLOMERATIVE CLUSTERING IN SPARSE ATOMIC DECOMPOSITIONS OF AUDIO SIGNALS

Bob L. Sturm, John J. Shynk, and Steffen Gauglitz

Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106-9560 USA

ABSTRACT

We present a correlation-based algorithm for the agglomerative clustering of atoms in sparse atomic decompositions of audio signals. Our goal is to demonstrate useful relationships between elements of the decomposition and the content of the original signal, for such purposes as analysis and modification. We evaluate the performance of the agglomeration algorithm using decompositions of synthetic and real audio signals, and discuss possible extensions of this work.

Index Terms— Clustering methods, signal analysis, signal resolution, time-frequency analysis.

1. INTRODUCTION

Methods for sparse atomic decomposition, such as Matching Pursuit (MP) [1], express a signal as a linear combination of components selected from an overcomplete and redundant set of finite-support unit-norm functions called *atoms* that collectively comprise a *dictionary*. After n iterations of MP, the representation of the size-M signal vector $\mathbf{x}$ can be expressed as

$$\mathbf{x} = \tilde{\mathbf{x}}(n) + \mathbf{r}(n) = \mathbf{G}(n)\mathbf{c}(n) + \mathbf{r}(n) \quad (1)$$

where the ith column $\mathbf{g}_i$ of the size $M \times n$ matrix $\mathbf{G}(n)$ is an atom selected from the size $M \times N$ dictionary $\mathbf{D}$, $\mathbf{c}(n)$ is a (column) vector of the decomposition coefficients, and $\mathbf{r}(n)$ is a residual signal vector. The nth-order approximation of $\mathbf{x}$ is a linear combination of n atoms selected from the columns of $\mathbf{D}$, i.e.,

$$\tilde{\mathbf{x}}(n) = c_0\mathbf{g}_0 + c_1\mathbf{g}_1 + \cdots + c_{n-1}\mathbf{g}_{n-1}. \quad (2)$$

Sparse atomic decompositions have been useful for many applications, such as the analysis of biomedical signals [2], blind source separation [3], and audio signal coding [4].

One can incorporate a diverse collection of atoms in the dictionary to describe a signal in a more sparse and meaningful way than using an orthogonal expansion [1]. For example, decomposing a signal using a dictionary of atoms of different scales results in a multiresolution representation, as opposed to the monoresolution representation of the short-term Fourier transform (STFT). Figure 1 shows examples of these two types of decompositions. In this paper, we use a dictionary of real-valued discrete Hann windows that are scaled, translated, and modulated, as follows:

$$g_l(m) = K_l w\Big(\frac{m-u_l}{s_l}\Big)\cos(\omega_l mT + \phi_l),\ 0 \le m \le M-1 \quad (3)$$

where $w(t)$ is the Hann window, T is the sampling period, and u_l, s_l, and ω_l are the translation, scaling, and modulation parameters, respectively. The coefficients $\{K_l\}$ are included to scale the discretized atoms to have unit norm.

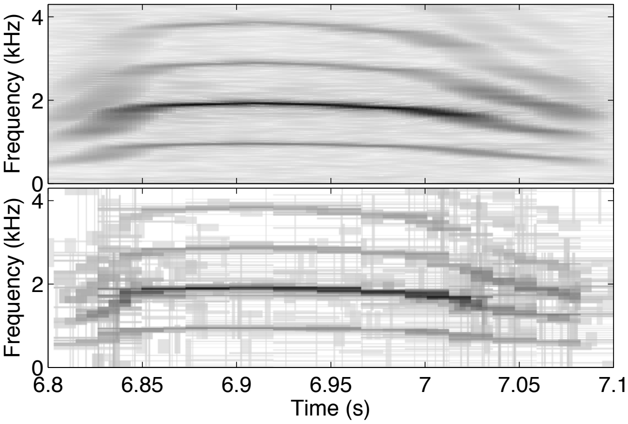

Fig. 1. Portion of a bird call signal in the time-frequency domain via the STFT (top), and a superposition of the time-frequency tiles of multiresolution atoms found by MP (bottom).

While the terms in a sparse decomposition of a signal may possess a higher level structure than those of an orthogonal decomposition, the significance of each atom with respect to each other, as well as relative to the content in the original signal, may not be obvious. For example, in a sparse decomposition of a musical signal, several atoms may characterize a single harmonic of one note. It is thus not straightforward how to work with high-level content of a musical signal (such as notes) via its sparse decomposition. Our goal here is to find and delimit structures in a sparse decomposition that correspond to specific content in a signal, such that we can rewrite (2) as a sum of k clusters ($k \ll n$) of weighted atoms, as follows:

$$\tilde{\mathbf{x}}(n) = \mathbf{m}_1(n) + \mathbf{m}_2(n) + \cdots + \mathbf{m}_k(n) \quad (4)$$

where each *molecule* $\mathbf{m}_i(n)$ is designed to have a clearer significance with respect to the content of the signal than do the individual atoms. We present an algorithm for building these molecules which, in contrast to molecular MP [5, 6], constructs them as a step subsequent to rather than during the decomposition. We anticipate that constructing molecules after the decomposition can provide many levels of resolution, and it imposes fewer restrictions on the process of the decomposition.

2. BUILDING MOLECULES

In our approach, each molecule in (4) is built by weighting and agglomerating the columns of $\mathbf{G}(n)$ that meet a minimum similarity criterion. One possible measure of similarity between two unit-norm

1-4244-1484-9/08/$25.00 ©2008 IEEE

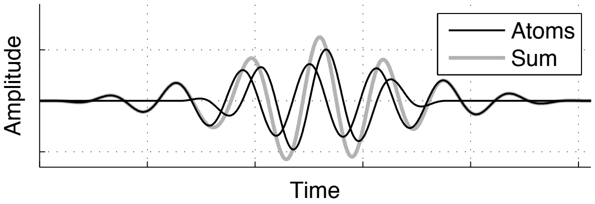

Fig. 2. Two real atoms that have the same modulation frequency and overlap in time, but differ in phase by $\pi/2$. The sum waveform is also shown.

atoms $\mathbf{g}_i$ and $\mathbf{g}_j$ is the magnitude of their cross-correlation:

$$r_{ij} \triangleq |\mathbf{g}_j^H \mathbf{g}_i| \tag{5}$$

where the superscript H denotes complex conjugate transpose. For the real-valued atoms in (3), this measure is obviously sensitive to their phases. Consider the two example atoms shown in Fig. 2 which possess the same modulation frequency. Even though we would judge these atoms to be similar, the phase lag of $\pi/2$ yields $r = 0.0001$, thus showing that (5) is not always a useful measure of similarity for real atoms overlapping in time and in frequency. In order to handle this drawback, the similarity measure can become phase invariant if the real atoms are first converted to analytic form. This is achieved for the atoms in (3) simply by replacing the cosine with a complex exponential. Converting the two atoms in Fig. 2 to analytic form gives the much higher similarity of $r = 0.87$.

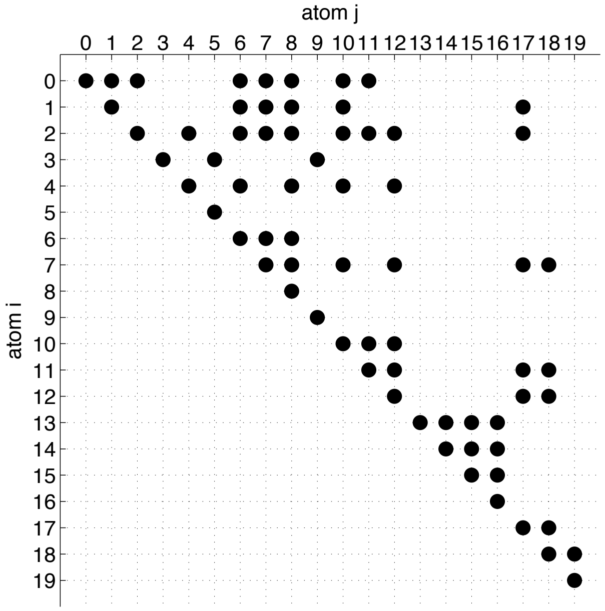

Fig. 3. Adjacency matrix $\mathbf{A}(n)$ denotes pairs of similar atoms (i.e., those that exceed the similarity threshold), indicated here by the dots.

2.1. Clustering Algorithm

Using an analytic form for the atoms, the similarity measure in (5) specifies that two atoms are similar when they sufficiently overlap in the time-frequency domain. In order to specify this sufficiency, define the correlation threshold $0 \leq \rho_{\min} \leq 1$ and construct the binary matrix $\mathbf{A}(n)$ with entries assigned as follows:

$$a_{ij} \triangleq \begin{cases} 1, & r_{ij} \geq \rho_{\min},\ 1 < j \leq n,\ i \leq j \\ 0, & \text{else.} \end{cases} \tag{6}$$

This upper-triangular *adjacency matrix* [6] specifies which pairs of atoms exceed the correlation threshold. By traversing the entries of $\mathbf{A}(n)$, the algorithm constructs molecules by agglomerating atoms that sufficiently overlap in time and in frequency.

Consider the adjacency matrix shown in Fig. 3. For the first molecule, the algorithm selects all atoms corresponding to the nonzero entries in the first row of $\mathbf{A}(n)$. These are weighted and added to form the first iteration of the first molecule: $\mathbf{m}_1(1) = c_0\mathbf{g}_0 + c_1\mathbf{g}_1 + c_2\mathbf{g}_2 + c_6\mathbf{g}_6 + \cdots + c_{11}\mathbf{g}_{11}$. Since $\mathbf{g}_0$ is similar to $\mathbf{g}_1$, the molecule is augmented with new atoms that are sufficiently similar to $\mathbf{g}_1$; these are represented by the nonzero entries in the second row of $\mathbf{A}(n)$. The second iteration of the first molecule thus produces $\mathbf{m}_1(2) = \mathbf{m}_1(1) + c_{17}\mathbf{g}_{17}$. Since $\mathbf{g}_0$ is also similar to $\mathbf{g}_2$, the molecule is augmented by all new atoms corresponding to the nonzero entries in the third row of $\mathbf{A}(n)$: $\mathbf{m}_1(3) = \mathbf{m}_1(2) + c_4\mathbf{g}_4 + c_{12}\mathbf{g}_{12}$. This process continues until no more new atoms are found, yielding the first completed molecule $\mathbf{m}_1(n)$. For the second molecule, the process begins with the first row of $\mathbf{A}(n)$ that is not included in the first molecule, which in this example is the fourth row. The molecule building process continues in this manner, and terminates when all rows of $\mathbf{A}(n)$ have been considered.

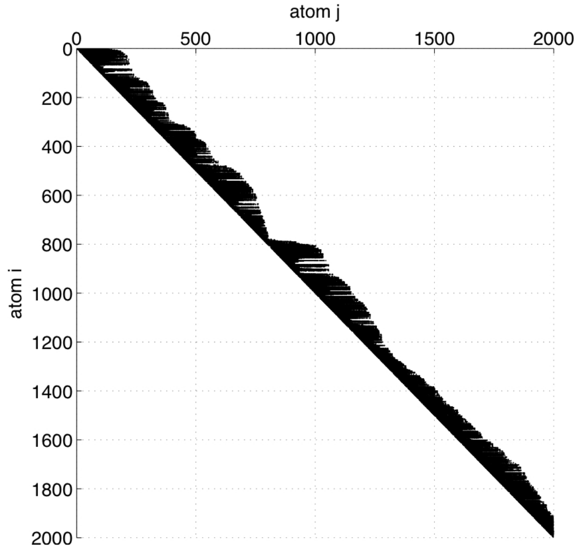

Fig. 4. Sorting the columns of $\mathbf{G}(n)$ according to the time positions of the atoms results in a very sparse adjacency matrix $\mathbf{A}(n)$.

2.2. Sliding Window Implementation

Sparse atomic decompositions of audio signals frequently result in very large numbers of atoms. Thus, it is generally impractical for the agglomeration algorithm to calculate and search $\mathbf{A}(n)$ for an entire decomposition. Since the clustering condition is based on the degree of time overlap, we should first permute the columns of $\mathbf{G}(n)$ based on the translation u_l of each atom before calculating $\mathbf{A}(n)$. For example, using the Hann atoms in (3), the columns of $\mathbf{G}(n)$ are permuted such that u_l increases with each column. This yields a very sparse adjacency matrix, as illustrated in Fig. 4. The algorithm may now use a sliding window approach where, for any given molecule, it looks only K atoms ahead of the first row included in the molecule. After each molecule is completed, the algorithm updates the adjacency matrix using the similarities between new atoms looking at only K atoms in each step. Obviously, the choice of the window size is important: if K is too small, molecules may be truncated; and if it is too large, the computational overhead may be excessive. However, using a small value for K may not necessarily be a problem if a post-processing step is used to agglomerate molecules that themselves sufficiently overlap in time and in frequency. This issue requires further study.

3. RESULTS FOR REAL AND SYNTHETIC SIGNALS

We decomposed several synthetic signals using an overcomplete dictionary of modulated Hann windows and the MP Toolkit [7]. The five test signals each consist of 36 Hann-windowed sinusoidal harmonic tones of duration 0.5 s in an equal-tempered scale built on 440 Hz. For each one of the tones, the amplitude of the fundamental and the phases of the harmonics are random variables. The harmonics have amplitudes that decay as A/k, $k = 2, \ldots, 10$, where A is the amplitude of the fundamental.

Ideally, the algorithm should find molecules in a decomposition corresponding to the individual harmonics. Each molecule should have a duration of 0.5 s, and since each test signal consists of 36 tones with 10 harmonics, the algorithm should find 360 such molecules. These results will depend, of course, on the value of the correlation threshold. As $\rho_{\min}$ increases, we expect to see an increase in the number of molecules because fewer pairs of atoms will exceed the threshold, and thus larger molecules will split into several smaller ones. This should also affect the mean duration of the molecules, because molecules with fewer atoms will tend to have shorter durations for these test signals.

Figure 5(a) shows the mean number of molecules found by the agglomeration algorithm from decompositions of the test signals for five different signal-to-residual ratios (SRRs), over a range of $\rho_{\min}$ values, and using $K = 500$ atoms. Figure 5(b) shows the corresponding mean duration of these molecules. For SRR ≈ 18 dB, the algorithm finds all the expected molecules with mean durations close to 0.5 s; listening to the synthesized molecules confirms this result. For SRR < 18 dB, the mean number of molecules is less than expected because several harmonics have yet to be extracted from the signal. Mean molecule durations are slightly longer than expected at high SRRs because some atoms selected by MP begin before or end after the corresponding tone. Both of these results are clearly dependent on whether real or analytic atoms are used to calculate the degree of similarity. This sensitivity suggests that the decompositions have many atoms that overlap in time and in frequency, and are out of phase, perhaps as a means to correct previous atoms.

We also constructed molecules from a decomposition of a bird call signal (11 s duration). For SRR $= 30$ dB, MP found $5,553$ real Hann atoms. The agglomeration algorithm employing analytic atoms with $\rho_{\min} = 0.1$ and $K = 500$ generated 274 molecules composed of ten or more atoms; these represent a total of $3,612$ atoms, which is about 65% of the original decomposition. Figure 6 shows nine such molecules built from this decomposition; observe that they correspond quite well to the harmonic structures of the original signal seen in Fig. 1. Some of these molecules can be agglomerated further as well, e.g., $\{2,6\}$, $\{3,8,9\}$, and $\{4,5,7\}$, to create larger structures.

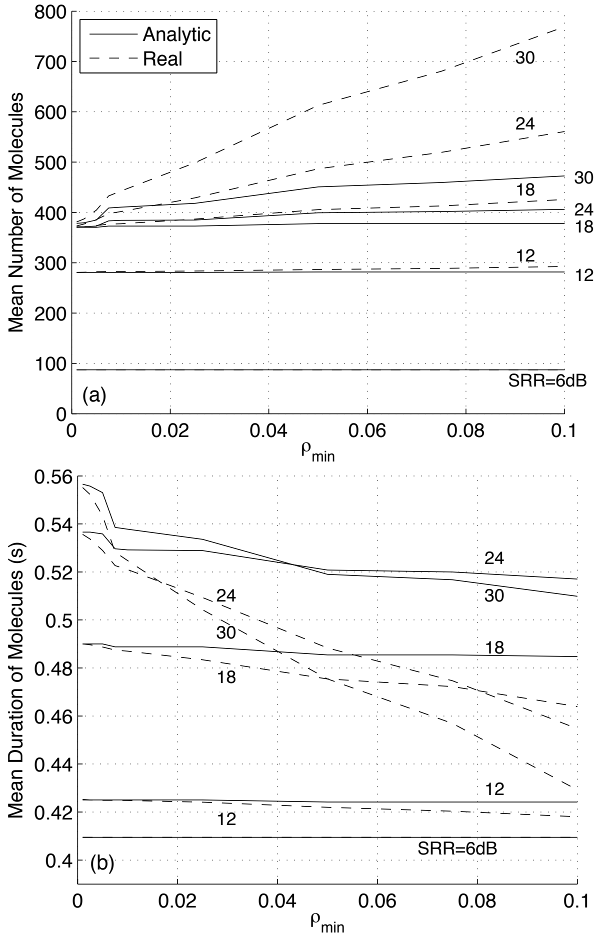

Fig. 5. Properties of molecules built from decompositions of synthetic signals as a function of the correlation threshold $\rho_{\min}$, the SRR, and the similarity measure r_{ij} in (5) calculated using real (dashed) and analytic (solid) atoms. (a) Mean number of molecules found. (b) Mean duration of molecules.

4. EXTENSIONS OF THE ALGORITHM

Due to the multiresolution nature of sparse atomic decompositions, several variations of the basic molecule building algorithm are possible. For example, agglomerative clustering could be performed for multiple SRRs, such that it might first build molecules at a low SRR, and then make them more precise by including atoms found at a higher SRR. Different sets of rules could also be specified for clustering depending on various properties of the atoms. For example, the algorithm might combine large-scale atoms differently than short-scale atoms since, in a musical signal for instance, atoms corresponding to transients will be related in a different way than atoms corresponding to the harmonic content.

A logical extension for the agglomeration algorithm is that it combine molecules according to frequency. For example, the molecules shown in Fig. 6 could be agglomerated to represent a complete “tweet” in the bird call signal. A lower-level representation could be created by segmenting the “tweet” into parts that signify the onset, sustain, and release of the sound. Such structurally meaningful representations can be useful for audio signal analysis, modeling, and transformation [8].

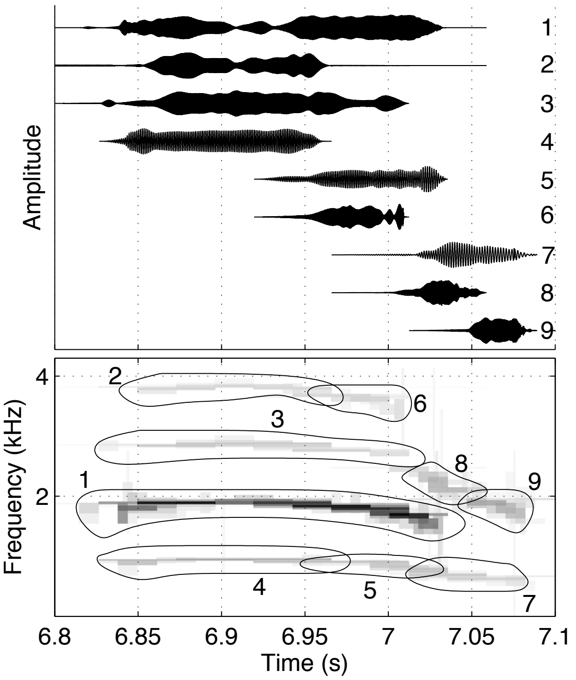

Fig. 6. A sparse decomposition of a bird call signal is processed by the agglomeration algorithm to create molecules corresponding to the harmonics of the signal, as seen in Fig. 1. Shown here are nine molecules in the time domain (top) and in the time-frequency domain (bottom).

In addition to providing an interface between the content of a signal and its sparse decomposition, molecule building could provide a way to prune a sparse decomposition without removing significant but low-energy structures in the signal. Figure 7 shows a histogram of the molecule sizes found in the decomposition of the bird call signal. Each bar from left to right in each cluster centered about an integer denotes $\rho_{\min}$ increasing linearly from 0.001 to 0.1. As $\rho_{\min}$ increases, the number of few-atom molecules increases while the number of the large molecules decreases. Even at low $\rho_{\min}$, many atoms in the decomposition are not sufficiently similar to create large molecules. Thus, a decomposition might be pruned or denoised by removing atoms that are not sufficiently similar.

5. CONCLUSION

We have presented, demonstrated, and evaluated an agglomerative clustering algorithm that makes more explicit the relationships between some content of a signal and the terms in its sparse atomic decomposition. The algorithm agglomerates similar atoms into molecules, where the similarity between two atoms is measured by the magnitude of their correlation (or that of their analytic equivalents). The end result is a set of molecules that provide an interface for working with specific content in a signal via its atomic decomposition at arbitrary levels of resolution. Experimental results demonstrate that the molecules correspond well with harmonic signal content, such as the harmonics of individual notes in a music signal. We also discussed various extensions and applications of this molecular approach, which we are currently investigating. Future work will also investigate the similarities and differences between this post-processing method of molecule building, and that done during the process of decomposition [5].

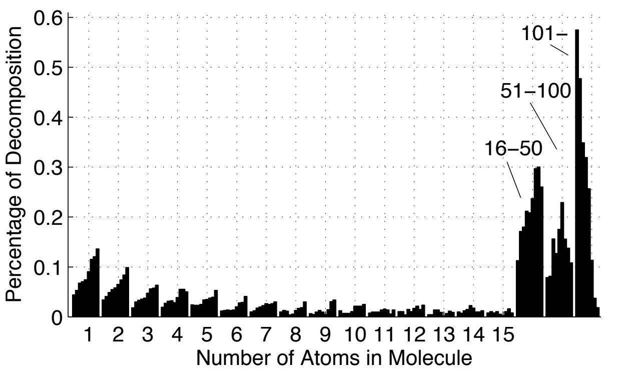

Fig. 7. Histogram of the number of molecules from the decomposition of the bird call signal, an excerpt of which is shown in Fig. 1. The molecules were built for SRR = 30 dB using analytic atoms to calculate the similarity measure.

6. ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under Grant CCF 0729229. We wish to thank Dr. Laurent Daudet, Dr. Pierre Leveau, and Dr. Curtis Roads for useful discussions regarding this work.

7. REFERENCES

[1] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[2] S. Krishnan, R. M. Rangayyan, G. D. Bell, and C. B. Frank, “Adaptive time-frequency analysis of knee joint vibroarthrographic signals for noninvasive screening of articular cartilage pathology,” *IEEE Trans. Biomed. Engineering*, vol. 47, no. 6, pp. 773–783, June 2000.

[3] P. Sugden and N. Canagarajah, “Underdetermined noisy blind separation using dual matching pursuits,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, vol. 5, Montreal, Quebec, Canada, May 2004, pp. 557–560.

[4] K. Vos, R. Vafin, R. Heusdens, and W. B. Kleijn., “High-quality consistent analysis-synthesis in sinusoidal coding,” in *Proc. Audio Eng. Soc. Int. Conf. High Quality Audio Coding*, Florence, Italy, Sep. 1999, pp. 244–250.

[5] L. Daudet, “Sparse and structured decompositions of signals with the molecular matching pursuit,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 5, pp. 1808–1816, Sept. 2006.

[6] P. Leveau and L. Daudet, “Multi-resolution partial tracking with modified matching pursuit,” in *Proc. European Signal Process. Conf.*, Florence, Italy, Sept. 2006.

[7] S. Krstulovic and R. Gribonval, “MPTK: Matching pursuit made tractable,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 3, Toulouse, France, May 2006, pp. 496–499.

[8] B. L. Sturm, L. Daudet, and C. Roads, “Pitch-shifting audio signals using sparse atomic approximations,” in *Proc. ACM Workshop Audio Music Comput. Multimedia*, Santa Barbara, CA, Oct. 2006, pp. 45–52.