A FAST ORTHOGONAL MATCHING PURSUIT ALGORITHM

Mohammad Gharavi-Alkhansari and Thomas S. Huang

Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign, Urbana, Illinois, U.S.A. 61801

ABSTRACT

The problem of optimal approximation of members of a vector space by a linear combination of members of a large overcomplete library of vectors is of importance in many areas including image and video coding, image analysis, control theory, and statistics. Finding the optimal solution in the general case is mathematically intractable. Matching pursuit, and its orthogonal version, provide greedy solutions to this problem. Orthogonal matching pursuit typically provides significantly better solution compared to the nonorthogonal version, but requires much more computation. This paper presents a fast algorithm for implementation of orthogonal matching pursuit which for many coding applications has computational complexity very close to that of the nonorthogonal version.

1. INTRODUCTION

Approximation of the members of a vector space by a linear combination of a small number of members in a possibly large set (dictionary) of redundant vectors in that space has been of interest in different areas of science. More formally, given a vector $\mathbf{x} \in R^M$, a scalar error threshold ε, and a set

$$U = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_P\} \subset R^M,$$

the problem is to find a subset

$$U^* = \{\mathbf{u}_{J_1}, \mathbf{u}_{J_2}, \ldots, \mathbf{u}_{J_S}\} \subset U, \quad 1 \leq J_1, J_2, \ldots, J_S \leq P$$

with smallest S, and a corresponding collection $\{\alpha_1, \alpha_2, \ldots, \alpha_S\}$, such that

$$\|(\alpha_1 \mathbf{u}_{J_1} + \alpha_2 \mathbf{u}_{J_2} + \cdots + \alpha_S \mathbf{u}_{J_S}) - \mathbf{x}\| \leq \varepsilon. \quad (1)$$

If we use the notation

$$\begin{aligned} \mathbf{U}^* &\overset{\text{def}}{=} \begin{bmatrix} \mathbf{u}_{J_1} & \mathbf{u}_{J_2} & \cdots & \mathbf{u}_{J_S} \end{bmatrix}_{M \times S}, \\ \boldsymbol{\alpha} &\overset{\text{def}}{=} \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_S \end{bmatrix}^T, \\ \hat{\mathbf{x}} &\overset{\text{def}}{=} \alpha_1 \mathbf{u}_{J_1} + \alpha_2 \mathbf{u}_{J_2} + \cdots + \alpha_S \mathbf{u}_{J_S} = \mathbf{U}^* \boldsymbol{\alpha}, \end{aligned}$$

then (1) may be written as $\|\mathbf{x} - \hat{\mathbf{x}}\| \leq \varepsilon$.

If the number S and the set U^* are found, the corresponding $\{\alpha_1, \alpha_2, \ldots, \alpha_S\}$ can simply be computed by least squares methods.

This problem is a difficult combinatorial optimization problem. In fact, it has recently been proven that, in the general case, finding the optimal solution is NP-hard [1]. However, an efficient suboptimal greedy solution to this problem has been discovered by different researchers in different contexts but with basically the same underlying mathematics.

In statistics, this greedy algorithm was found for the computation of conditional expectation of random variables and was named *projection pursuit* [2]. In control theory, such a method was developed for nonlinear system identification [3]. In the context of time-frequency decomposition, it was named *matching pursuit* [4] and was used in signal analysis for extraction of patterns from noisy signals. In the context of image coding, it was developed for a generalized image coding method unifying transform coding, vector quantization, and fractal coding [5].

The orthogonal matching pursuit was developed independently by Chen et al. [3], Pati et al. [6], Davis et al. [7], and Gharavi-Alkhansari and Huang [8]. Rate-distortion optimized versions of matching pursuit were proposed in [9], [10], and [11].

Vetterli and Kalker used rate-distortion optimized matching pursuit for motion compensated video coding [11]. Neff and Zakhor used matching pursuit for coding motion residual images of video sequences [12, 13]. Gharavi-Alkhansari and Huang used rate-distortion optimized matching pursuit for generalized fractal image and video coding [9, 10].

The essence of matching pursuit is that, for a given vector $\mathbf{x}$ to be approximated, first choose the vector from the dictionary on which $\mathbf{x}$ has the longest projection. Then, remove any component of the form of the selected vector from $\mathbf{x}$, i.e., orthogonalize $\mathbf{x}$ with respect to the selected dictionary vector, and obtain the residual of $\mathbf{x}$. The selected dictionary vector is in fact the one that results in the residual of $\mathbf{x}$ with the

```
for $i := 1$ to $P$
    $u_i := \|\mathbf{u}_i\|$
$T := \{1, 2, \ldots, P\}$
$k := 0$
while $\|\mathbf{x}\| > \varepsilon$ {
    $k := k + 1$
    for all $i \in T$
        $c_i := \frac{1}{u_i}(\mathbf{x} \cdot \mathbf{u}_i)$
    find $J_k$ such that $c_{J_k} = \max_{i \in T} c_i$
    $T := T - \{J_k\}$
    $r_{k,J_k} := u_{J_k}$
    $\mathbf{u}_{J_k} := \frac{1}{u_{J_k}} \mathbf{u}_{J_k}$
    $\mathbf{x} := \mathbf{x} - c_{J_k} \mathbf{u}_{J_k}$
    for all $i \in T$ {
        $r_{k,i} := \mathbf{u}_i \cdot \mathbf{u}_{J_k}$
        $\mathbf{u}_i := \mathbf{u}_i - r_{k,i} \mathbf{u}_{J_k}$
        $u_i := \|\mathbf{u}_i\|$ } }
$S := k$
$\boldsymbol{\alpha} := \left(\mathbf{U}^{*T}\mathbf{U}^{*}\right)^{-1}\mathbf{U}^{*T}\mathbf{x}$
```

Figure 1: Basic Algorithm for orthogonal matching pursuit.

smallest energy. Repeat this process for the residual of $\mathbf{x}$ with the rest of dictionary vectors until the norm of the residual becomes smaller than the threshold ε.

In matching pursuit, after a vector in the dictionary is selected, one may remove any component of its form not only from $\mathbf{x}$, but also from all other dictionary vectors before repeating the process. This version of the method is called *orthogonal matching pursuit* and is computationally more expensive than the nonorthogonal version, but typically gives significantly better results in the context of coding. The basic orthogonal matching pursuit algorithm, is shown in Figure 1.

2. PROPOSED ALGORITHM

In this paper, we assume that $1 \leq S \ll M < P$ and that all the computations, including all inner products, must be done in real time and cannot be precomputed, as it is the case in many applications of this algorithm in image and video coding [5, 11, 8, 9, 10]. With this assumption, most of the computation time for the algorithm of Figure 1 is spent on the vector operations. For this algorithm, under the above assumption, and ignoring lower order terms, approximately $(8S+2)MP$ arithmetic computations are required.

The proposed algorithm is shown in Figure 2 and requires approximately $(2S+2)MP$ arithmetic operations, which is 2.5 to 4 times faster than the algorithm of Figure 1. This is the same as the $(2S+2)MP$ arithmetic operations required for the standard (nonorthogonal) matching pursuit. This speed up is made possible by the following observations.

1) If we use the notation $x \overset{\text{def}}{=} \|\mathbf{x}\|$, then due to the orthogonalization process, at stage k, u_i and x can be updated recursively using the following updating formulas, and without directly using $\mathbf{u}_i$ and $\mathbf{x}$:

$$x = \sqrt{x^2 - c_{J_k}^2} \tag{2}$$

$$u_i = \sqrt{u_i^2 - r_{k,i}^2}. \tag{3}$$

2) Lemma: Let $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ be three vectors. Let us write each of $\mathbf{a}$ and $\mathbf{b}$ in the form of sum of two vectors:

$$\mathbf{a} = \mathbf{a}_1 + \mathbf{a}_2 \tag{4}$$

$$\mathbf{b} = \mathbf{b}_1 + \mathbf{b}_2, \tag{5}$$

where $\mathbf{a}_1$ and $\mathbf{b}_1$ are orthogonal to $\mathbf{c}$, and $\mathbf{a}_2$ and $\mathbf{b}_2$ are along the direction of $\mathbf{c}$. Then

$$\mathbf{a}_1 \cdot \mathbf{b}_1 = \mathbf{a} \cdot \mathbf{b} - \|\mathbf{a}_2\| \; \|\mathbf{b}_2\|.$$

Using this lemma, in Figure 1, at each iteration the new c_i can be computed from the old c_i of the previous iteration. This, with the new method of updating x in item 1 above, completely removes the need for updating $\mathbf{x}$, and the computation $\mathbf{x} := \mathbf{x} - c_{J_k}\mathbf{u}_{J_k}$ can be avoided.

By a similar argument, but with more involved mathematics, at each iteration, new $r_{k,i}$ can also be computed from old values of $r_{k,i}$ of previous iterations and from inner product of original $\mathbf{u}_i$'s (computed as necessary). Then $\mathbf{u}_i$'s need not be updated and the line $\mathbf{u}_i := \mathbf{u}_i - r_{k,i}\mathbf{u}_{J_k}$ in Figure 1 can be removed.

The above two techniques remove the bulk of computation from the algorithm. Further optimizations are also implemented in the algorithm of Figure 2, which affect the lower order terms, and are also important when the assumption $1 \leq S \ll M < P$ is not held. The following is the most significant of these optimization.

3) The last line of the algorithm of Figure 1 finds the coefficients vector $\boldsymbol{\alpha}$ using least squares. The least squares computation can be done by a QR decomposition [14] of the matrix $\mathbf{U}^*$, i.e., by writing

$$\mathbf{U}^*{}_{M\times S} = \mathbf{Q}_{M\times S}\mathbf{R}_{S\times S},$$

where the columns of $\mathbf{Q}$ are orthonormal and $\mathbf{R}$ is an upper-triangular matrix. Then, $\boldsymbol{\alpha}$ can be found [14, page 238] from

$$\boldsymbol{\alpha} = \mathbf{R}^{-1}\mathbf{Q}^T\mathbf{x}.$$

The upper triangular matrix $\mathbf{R}$ and the vector $\mathbf{Q}^T\mathbf{x}$ are computed directly in the process of selecting the library vectors in Figure 2:

$$\mathbf{R} = \begin{bmatrix} r_{1,J_1} & r_{1,J_2} & \cdots & r_{1,J_S} \\ & r_{2,J_2} & \cdots & r_{2,J_S} \\ & & \ddots & \vdots \\ 0 & & & r_{S,J_S} \end{bmatrix} \quad (6)$$

$$\mathbf{p} \overset{\text{def}}{=} \mathbf{Q}^T\mathbf{x} = \begin{bmatrix} c_{J_1} & c_{J_2} & \cdots & c_{J_S} \end{bmatrix}^T \quad (7)$$

As $\mathbf{R}$ is an upper triangular matrix, the computation of $\mathbf{R}^{-1}\mathbf{p}$ in Figure 2 needs far less computation that inversion of a general $S \times S$ matrix and can be done by *back substitution*:

```
for $i := S$ downto 1  {
    for $j := S$ downto $i + 1$
        $c_i := c_i - r_{i,J_j} c_j$
    $c_i := c_i / r_{i,J_i}$  }
```

Therefore, $\boldsymbol{\alpha}$ can be found with a small number of computations after the library vectors are selected.

The columns of matrix $\mathbf{Q}$ are columns of matrix $\mathbf{U}^*$ orthonormalized using the Gram-Schmidt algorithm. While elements of $\boldsymbol{\alpha}$ are the coefficients of columns of $\mathbf{U}^*$ that can generate an approximation of $\mathbf{x}$, elements of $\mathbf{p}$ are the coefficients of columns of $\mathbf{Q}$, i.e.,

$$\hat{\mathbf{x}} = \mathbf{U}^*\boldsymbol{\alpha} = \mathbf{Q}\mathbf{p}$$

and $\mathbf{R}$ is the transformation that can convert $\boldsymbol{\alpha}$ to $\mathbf{p}$, i.e.,

$$\mathbf{p} = \mathbf{R}\boldsymbol{\alpha}.$$

3. ACKNOWLEDGMENTS

This work was supported in part, by Joint Services Electronics Program Grant ONR N00014-96-1-0129; and, in part, by a grant from Mitsubishi Electric.

4. REFERENCES

[1] G. Davis, *Adaptive Nonlinear Approximations*. PhD thesis, New York University, New York, NY, Sept. 1994.

[2] J. H. Friedman and W. Stuetzle, "Projection pursuit regression," *Journal of the American Statistical Association*, vol. 76, pp. 817–823, Dec. 1981.

[3] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *International Journal of Control*, vol. 50, no. 5, pp. 1873–1896, 1989.

[4] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3397–3415, Dec. 1993.

[5] M. Gharavi-Alkhansari and T. S. Huang, "Fractal-based techniques for a generalized image coding method," in *Proceedings of IEEE International Conference on Image Processing*, vol. 3, (Austin, TX), pp. 122–126, Nov. 13–16, 1994.

[6] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal projection pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proceedings of Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, vol. 1, (Pacific Grove, CA), pp. 40–44, Nov. 1–3, 1993.

[7] G. Davis, S. Mallat, and Z. Zhang, "Adaptive time-frequency decompositions," *Optical Engineering*, vol. 33, pp. 2183–2191, July 1994.

[8] M. Gharavi-Alkhansari and T. S. Huang, "Generalized image coding using fractal-based methods," in *Proceedings of the International Picture Coding Symposium*, (Sacramento, CA), pp. 440–443, Sept. 21–23, 1994.

[9] M. Gharavi-Alkhansari and T. S. Huang, "Fractal image coding using rate-distortion optimized matching pursuit," in *Proceedings of the SPIE, Visual Communications and Image Processing*, vol. 2727, (Orlando, FL), pp. 1386–1393, Mar. 17–20, 1996.

[10] M. Gharavi-Alkhansari and T. S. Huang, "Fractal video coding by matching pursuit," in *Proceedings of IEEE International Conference on Image Processing*, vol. 1, (Lausanne, Switzerland), pp. 157–160, Sept. 16–19, 1996.

[11] M. Vetterli and T. Kalker, "Matching pursuit for compression and application to motion compensated video coding," in *Proceedings of IEEE International Conference on Image Processing*, vol. 1, (Austin, TX), pp. 725–729, Nov. 13–16, 1994.

[12] R. Neff, A. Zakhor, and M. Vetterli, "Very low bit rate video coding using matching pursuits," in *Proceedings of the SPIE, Visual Communications and Image Processing I*, vol. 2308, (Chicago, IL), pp. 47–60, Sept. 25–28, 1994.

[13] R. Neff and A. Zakhor, "Matching pursuit video coding at very low bit rates," in *DCC'95: Data Compression Conference*, (Snowbird, UT), Mar. 28–30, 1995.

[14] S. J. Leon, *Linear Algebra with Applications*. New York: Macmillan Publishing Company, 3rd ed., 1990.

```
$x := \|\mathbf{x}\|$
if $x \le \varepsilon$ {
    $S := 0$
    return }
$T := \{1, 2, \ldots, P\}$
for $i := 1$ to $P$ {
    $e_i := \mathbf{u_i} \cdot \mathbf{u_i}$
    $u_i := \sqrt{e_i}$
    $c_i := \frac{1}{u_i}(\mathbf{x} \cdot \mathbf{u}_i)$ }
$k := 1$
find $J_k$ such that $c_{J_k} = \max_{i \in T} c_i$
$T := T - \{J_k\}$
$x := \sqrt{x^2 - c_{J_k}^2}$
while $x > \varepsilon$ {
    $r_{k,J_k} := u_{J_k}$
    for all $i \in T$ {
        $r_{k,i} := \frac{1}{u_{J_k}}\left(\mathbf{u}_{J_k} \cdot \mathbf{u}_i - \sum_{j=1}^{k-1} (r_{j,J_k}\ r_{j,i})\right)$
        $c_i := c_i u_i - c_{J_k} r_{k,i}$
        $e_i := e_i - r_{k,i}^2$
        $u_i := \sqrt{e_i}$
        $c_i := c_i / u_i$ }
    $k := k + 1$
    find $J_k$ such that $c_{J_k} = \max_{i \in T} c_i$
    $T := T - \{J_k\}$
    $x := \sqrt{x^2 - c_{J_k}^2}$ }
$S := k$
$\boldsymbol{\alpha} := \begin{bmatrix} r_{1,J_1} & r_{1,J_2} & \cdots & r_{1,J_S} \\ & r_{2,J_2} & \cdots & r_{2,J_S} \\ & & \ddots & \vdots \\ 0 & & & r_{S,J_S} \end{bmatrix}^{-1} \begin{bmatrix} c_{J_1} \\ c_{J_2} \\ \vdots \\ c_{J_S} \end{bmatrix}$
```

Figure 2: The proposed fast algorithm for orthogonal matching pursuit.