IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, 26 Sept.-2 Oct. 2020, Waikoloa, HI, USA
DOI: 10.1109/IGARSS39084.2020.9323532

RAPID ESTIMATION OF ORTHOGONAL MATCHING PURSUIT REPRESENTATION

Ayan Chatterjee, Peter W. T. Yuen

Centre for Electronic Warfare, Information and Cyber (CEWIC), Cranfield Defence and Security, Cranfield University, Defence Academy of the United Kingdom, Shrivenham, SN6 8LA, UK.

ABSTRACT

Orthogonal Matching Pursuit (OMP) has proven itself to be a significant algorithm in image and signal processing domain in the last decade to estimate sparse representations in dictionary learning. Over the years, efforts to speed up the OMP algorithm for the same accuracy has been through variants like generalized OMP (g-OMP) and fast OMP (f-OMP). All of these algorithms solve OMP recursively for each signal sample among 'S' number of samples. The proposed rapid OMP (r-OMP) runs the loop for 'N' atoms, simultaneously estimating for all samples, and, in a real scene since N $\ll$ S, the proposed approach speeds up OMP by several orders of magnitude. Experiment on a real scene with a popular dictionary learning algorithm, K-SVD, show that the proposed r-OMP completes K-SVD in $\approx$4% of the computational time compared to using OMP.

Index Terms— OMP, sparse representation, basis, KSVD

1. INTRODUCTION

Dictionary learning is an approach to decomposition of a multi-dimensional hyperspectral image matrix into two matrices: one containing a collection of 'n' dimensional vectors in a 'b' dimensional hyperspectral image space (n $<$ b) called dictionary and the other matrix contains representation (or the magnitude of contribution) of the dictionary entities for each pixel. The 'n' dimensional vectors (similar to unit vectors in higher dimensional space) are independent of one another, which are obtained by various mathematical optimisation methods, and each symbolises a unique direction in the presence of measurement noise.

Dictionary learning (DL) is a process to decompose or split up compressible signals into basis functions called atoms in a dictionary (D), and a sparse matrix called representation (a). For a signal sample/pixel 'y', the DL decomposition is mathematically expressed as:

$$y = Da \tag{1}$$

This work was supported in part by The Defence Science and Technology Laboratory (DSTL) under project number DSTLX-1000103251. The source code of the proposed algorithm is available at https://www.doi.org/10.24433/CO.6785856.v1.

One of the ways to solve this inverse problem is to first establish the representation 'a' and then update the 'D', as used by sparse coding approaches for anomaly and target detection applications [1], sub-sampling with k-means classification in [2] and with spectral angle measurements in [3], to the popular generalized k-means or K-SVD [4] algorithm. The estimation of 'a' is either done through convex relaxation approaches minimizing for ℓ_p norm (p $>$ 0, most algorithms follow p = 2) [5, 6], or through greedy methods like the popular Orthogonal Matching Pursuit (OMP) with some recent uses in [7, 8, 9, 10].

Several variants of OMP exist in efforts to speed up the algorithm, like, the generalized OMP (g-OMP) [11] selecting multiple atoms at the same time based on descending order of atomic correlation with the selected signal sample. Other variants like fast OMP (f-OMP) [12], Cholesky OMP, and batch OMP in [13] using Cholesky or QR update process to reduce the computation of the dictionary inverse. The common factor between all these variants of OMP is that OMP is estimated on *per-signal case*, imposing a massive computational cost in terms of execution time for large datasets like AVIRIS hyperspectral images [14]. The rapid OMP proposes to iterate over *per-atom case* to achieve massive improvements in computational time while keeping the OMP algorithm's procedure intact.

2. ORTHOGONAL MATCHING PURSUIT

Orthogonal Matching Pursuit (OMP) approximates sparse representation for a signal iteratively selecting one dictionary atom at a time that is most correlated with the signal's residue. The correlation of two vectors is given by:

$$C(a,b) = \frac{a^T}{||a||_2}\frac{b}{||b||_2} \tag{2}$$

In DL, the elements of 'D' are normalised, and the atom with maximum correlation can be found by seeking for max($|D^T r|$) since the position of atom with max($|D^T \frac{r}{||r||_2}|$) is also with max($|D^T r|$) . Using this property, OMP achieves its representation in 2 sub-steps (algorithm 1) for a given iteration. OMP loops untill the residue is within error tolerance:

© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

- The first sub-step is to estimate the correlation of the atoms with the signal residue 'r', followed by selection of the atom with maximum correlation i.e. atom position with $\max(|D^T r|)$. The selected atom gets added to the selected subset of atoms from previous iterations.

- The second sub-step is to estimate the representation of with the selected atoms (usually by least squares) and update the residue by $r = y - Da$.

Algorithm 1 Orthogonal Matching Pursuit (OMP)

Input signal samples/pixel 'Y', dictionary 'D', maximum number of atoms per sample/pixel 'm', and 'n' noise/error tolerance.
Let 'S' be the total number of signal samples/pixels, and representation matrix 'a' be be a zeros matrix with dimension 'N' number of atoms in D by S
for j = 1 to S **do** ▷ iterate for all samples
    r_j = Y_j
    Let $k = 1$ ▷ inner loop counter
    while $k \leq m$ AND $\sum|r_j| \leq n$ **do**
        $\phi_j = |D^T r_j|$
        $p_k \leftarrow \max(\phi_k)$ ▷ position of maximum
        $p_j \leftarrow \{p_j, p_k\}$
        $D_j = D(p_j)$ ▷ subset D to matched atoms
        $a(p_j, j) = (D_j^T D_j)^{-1} D_j^T y_j$ ▷ by least squares
        $r_j = y_j - Da_j$ ▷ update residue
        $k = k + 1$
Output representation 'a'

3. PROPOSED RAPID OMP

OMP algorithms are solved by iterating for each signal sample, in two substeps. In scenes like AVIRIS images, the number of samples (S) can be in millions, thus, increasing the computational time drastically by dictionary learning algorithms using OMP for representation. Realising that the number of signal samples is far greater than the number of dictionary atoms, the proposed rapid OMP (r-OMP) iterates over the number of atoms, approaching atom selection and residue update the following way:

3.1. Atom selection

The atom selection step is to find out which atom has the most correlation with each signal. While the absolute value of $D^T y$ (say ϕ) is a matrix operation, the selection criteria for the maximum value require a different approach in order to avoid a loop for all samples. For each atom in a loop with counter k (1$\leq$ k $\leq$N), r-OMP finds if it has the maximum value for all the samples with

$$V_{k,y} = \begin{cases} = 1 & \text{when } \sum_j^N (\phi_j - \phi_k > 0) = 0 \\ = 0 & \text{when } \sum_j^N (\phi_j - \phi_k > 0) \neq 0 \end{cases} \quad (3)$$

The samples for which kth atom is with the highest residue will have no positive value in $\sum_N(\phi - \phi_k)$ and the expression $V_{k,y}$ will return a binary Boolean value for kth atom and signal sample y. The constituents of V are

$$V_{k,y} = \begin{cases} = 1 & \text{atom selected for representation} \\ = 0 & \text{atom not selected for representation} \end{cases} \quad (4)$$

3.2. Estimating the representation and residue update

The next step of OMP is to estimate the representation for the selected atoms. The inverse of orthonormal with ℓ_2 norm basis is a simple transpose operation such as a complete dictionary with PCA basis. The representation update stops at this stage for orthonormal atoms, however, the basis for overcomplete sparse representations is rarely orthonormal. The inverse is estimated by least squares function in equation (5) where D_y refers to the dictionary subset that is selected by OMP and y is the signal sample.

$$\text{representation } a = \underbrace{(D_y^T D_y)^{-1}}_{\alpha} \underbrace{D_y^T y}_{\beta} \quad (5)$$

The computational cost is dependent on how the dictionary 'D' is inverted. For a D matrix with a size of 'b' bands by 'N' atoms, the inversion of D can be done by the following ways:

- Using Cholesky method by splitting up the D to a lower triangular matrix (say L) and its conjugate transpose (L^T). This method has a drawback in handling near-singular and rank deficient D and has a complexity of $O(bN^2)$,

- With QR decomposition where the D is split into an orthonormal matrix Q and an upper triangular matrix R. This approach is more stable than Cholesky and has a complexity of $O(N^3)$, and

- by Moore–Penrose inverse decomposing the D by Singular Value Decomposition(SVD). This method is the most stable method among the three approaches and has a complexity of $O(N^3 + bN^2) \approx O(N^3)$ when $N \gg b$.

Each sample 'y' has fractional contribution of a few atoms (say $D_s, D_s \subset D$), and, OMP computes the inverse of 'D_s' for a total of 'S' samples by use of iterative loops has a cost of $O(SN^3)$. A natural scene contains groups of similar materials whose membership to the 'D' in terms of atom selection and pseudo-norm is the same, even though the magnitude of representation may be different, i.e., with different ℓ_p norm,

where p > 0, but same ℓ_0 norm with similar atom selection. The proposed rapid OMP computes the inverse of D_s for for all groups once ('α' in equation (5)), and for 'K' groups the total cost is O(KN^3), S $\gg$ K. Additionally, the speed improvements seen is in the efficiency in matrix computations and the inefficiency in a native for-loop with modern languages.

There are multiple possibilities to implement the solution to incorporate V on different programming languages. However, the estimation of α in equation (5) requires an additional loop for each unique collection of atoms in $V_{k,y}$. This can be estimated easily through methods existing in common programming languages like in MATLAB [15] and through NumPy in Python [16]. The complexity of the algorithm increases with increase in the combination possibilities of material mixtures, but, in increase in computational complexity per iteration is justified in real hyperspectral images where the number of signal samples are often >1000x more than the number of dictionary atoms. The algorithm estimates the representation for OMP by this approach, and estimating the residue by

$$\text{residue } r_y = y - D\alpha_y\beta_y \tag{6}$$

The entire process is written down in algorithm 2.

Algorithm 2 Proposed rapid OMP (r-OMP)

Input signal samples/pixel 'Y', dictionary 'D', maximum number of atoms per sample/pixel 'm', and 'n' noise/error tolerance.
Let representation 'a' and binary matrix V be two zeros matrix with dimensions of number of atoms (N) by number of samples/pixels (S).
Initialise with $r_s = Y$ and Y_s be all pixels in the image.
for j = 1 to m **do**
    $\phi = |D^T r_s|$
    for k = 1 to N **do** ▷ for all atoms
        Estimate V_k from equation (3)
    $\beta_e = D^T Y_s$ ▷ from equation (5)
    $W_s, p_s \leftarrow$ unique(V) ▷ unique columns W_s and position/index p_s
    Let w be the total number of elements in W_s
    for e = 1 to w **do**
        $D_e \subset D$
        $a_e = (D_e^T D_e)^{-1}\beta_e$
    $a \leftarrow a \bigcup a_e$ ▷ Update representation values for 'e' positions
    r = Y - Da ▷ equation (6)
    $Y_s \subset Y$ ▷ subset samples whose $\sum|r_y| > n$
    $r_s \subset r$ ▷ subset to residue of Y_s
Output representation 'a'

4. RESULTS

The results presented are run on MATLAB R2019b on a Windows 10 version 1909 PC with Intel i7-6700K 4GHz processor and 32GB of RAM. The first result presents a synthetic run with 10,000 (or 10k), 50k, and 100k samples with 30 dictionary atoms, presented with OMP, f-OMP, g-OMP and the proposed r-OMP. The experiment in figure 1 shows speed advances with the proposed r-OMP, completing in ≈15%, 7%, and 1.3% of the time taken by f-OMP, g-OMP, and OMP respectively.

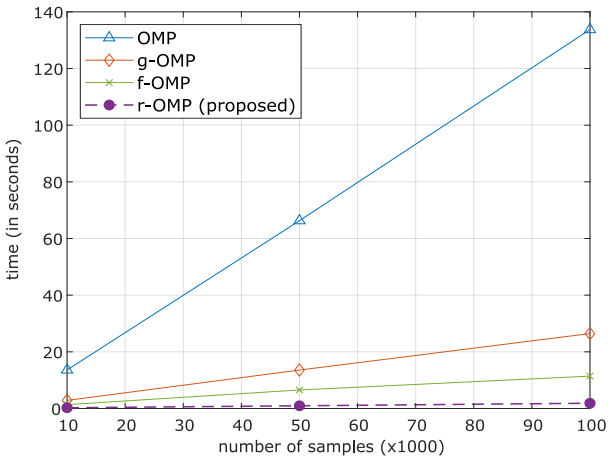


Fig. 1: Time taken by variants of OMP

While synthetic experiments provide some conclusion, it is crucial to present the impact on a real scene. An experiment presented with the popular dictionary learning algorithm, K-SVD [4], which uses OMP for representation, is tested on the scene H23Dual, which is part of Selene dataset [17]. Natural materials like grass, soil and tree cover over 95% of Selene scene, and artificial materials such as ground markers, path, concrete, building and coloured panels cover the remaining scene. H23Dual was registered from images taken by HySpex VNIR-1600 and SWIR-384 sensors and contain 1876 lines with 384 samples (or ≈0.7 million pixels), and 448 spectral bands (from 0.4 to 2.5 µm). QUAC [18] was applied to the raw data using ENVI software with a generic sensor to obtain the reflectance of this dataset.

K-SVD is run on the H23Dual image with 10, 20, and 30 atoms for a maximum of three atoms per pixel, and 50 training iterations. Results presented in table 1 show that r-OMP completed in ≈half hour each compared to OMP in ≈13.5 hours for the same mean error of ≈3 in Manhattan distance on an average. The proposed r-OMP has the potential for further acceleration by the use of parallel processing and GPU.

Table 1: Time taken (in minutes) by K-SVD dictionary learning on Selene H23Dual scene for 50 iterations

K-SVD with	OMP (existing)	r-OMP (proposed)
10 atoms	811.19	27.53
20 atoms	824.33	31.31
30 atoms	811.89	39.15
Average time taken	815.80 ± 7.39	32.66 ± 5.93

5. CONCLUSION

OMP is a significant algorithm in sparse dictionary learning/compressed sensing, and any enhancement to the algorithm is crucial to the research domain. r-OMP proposes enhancements to speed up the OMP algorithm by several orders of magnitude. The strength of r-OMP lies in that the number of samples in a scene are often very large compared to the number of overcomplete basis vectors. The enhancement is evidenced from Selene H23Dual scene which learns K-SVD dictionary atoms in $\approx$4% of the time as compared with the original OMP, or, $\approx$25 times faster for that scene.

6. REFERENCES

[1] A. Chatterjee and P. Yuen, "Sample selection with somp for robust basis recovery in sparse coding dictionary learning," *IEEE Letters of the Computer Society*, 2019.

[2] A. Chatterjee and P. W. T. Yuen, "Endmember learning with k-means through scd model in hyperspectral scene reconstructions," *Journal of Imaging*, vol. 5, no. 11, 2019.

[3] J. Bowles, D. Gillis, and P. Palmadesso, "New improvements in the orasis algorithm," in *2000 IEEE Aerospace Conference. Proceedings (Cat. No.00TH8484)*, vol. 3, March 2000, pp. 293–298 vol.3.

[4] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, Nov 2006.

[5] M. S. Asif and J. Romberg, "Fast and accurate algorithms for re-weighted ℓ_1-norm minimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 23, pp. 5905–5916, Dec 2013.

[6] V. Papyan, J. Sulam, and M. Elad, "Working locally thinking globally: Theoretical guarantees for convolutional sparse coding," *IEEE Transactions on Signal Processing*, vol. 65, no. 21, pp. 5687–5701, Nov 2017.

[7] Y. Chen, G. Li, and Y. Gu, "Active orthogonal matching pursuit for sparse subspace clustering," *IEEE Signal Processing Letters*, vol. 25, no. 2, pp. 164–168, Feb 2018.

[8] J. Li, M. Li, X. Yao, and H. Wang, "An adaptive randomized orthogonal matching pursuit algorithm with sliding window for rolling bearing fault diagnosis," *IEEE Access*, vol. 6, pp. 41 107–41 117, 2018.

[9] S. Zhuang, W. Zhao, R. Wang, Q. Wang, and S. Huang, "New measurement algorithm for supraharmonics based on multiple measurement vectors model and orthogonal matching pursuit," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 6, pp. 1671–1679, June 2019.

[10] B. Ng and L. Rosenberg, "Sparse detection in seaclutter using orthogonal matching pursuit," in *2019 IEEE Radar Conference (RadarConf)*, April 2019, pp. 1–6.

[11] J. Wang, S. Kwon, and B. Shim, "Generalized orthogonal matching pursuit," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6202–6216, Dec 2012.

[12] M. Yaghoobi, D. Wu, and M. E. Davies, "Fast nonnegative orthogonal matching pursuit," *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1229–1233, Sep. 2015.

[13] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit," *CS Technion*, vol. 40, 01 2008.

[14] *AVIRIS*, (accessed December 25, 2019). [Online]. Available: https://aviris.jpl.nasa.gov

[15] *Unique function in MATLAB*, (accessed December 25, 2019). [Online]. Available: https://uk.mathworks.com/help/matlab/ref/unique.html#d117e1554095

[16] *Unique function in Python*, (accessed December 25, 2019). [Online]. Available: https://docs.scipy.org/doc/numpy/reference/generated/numpy.unique.html

[17] J. Piper, "A new dataset for analysis of hyperspectral target detection performance," in *HSI 2014, Hyperspectral Imaging and Applications Conference, October 2014,Coventry UK.*, October 2014.

[18] B. G. S. M. A.-G. Lawrence S. Bernstein, Xuemin Jin, "Quick atmospheric correction code: algorithm description and recent upgrades," *Optical Engineering*, vol. 51, no. 11, pp. 1 – 12 – 12, 2012.



https://dspace.org/

Rapid estimation of orthogonal matching pursuit representation

Chatterjee, Ayan; Yuen, Peter W. T.

2021-02-17
IEEE
https://doi.org/10.1109/IGARSS39084.2020.9323532
Chatterjee A, Yuen P. (2021) Rapid estimation of orthogonal matching pursuit representation. In: IGARSS 2020: International Geoscience and Remote Sensing Symposium, 26 September - 2 October 2020, Waikoloa, HI, USA

Downloaded from a DSpace repository. https://dspace.org/