Identifying Archetypes by Exploiting Sparsity of Convex Representations

V. Abrol, P. Sharma, A. K Sao
SCEE, IIT Mandi, India
Email: {vinayak_abrol,pulkit_s}@students.iitmandi.ac.in, anil.iitmandi.ac.in

***Abstract*—This paper presents a computationally efficient greedy archetypal analysis (GAA) algorithm. GAA leverages the underlying sparseness property of AA, and thus is scalable to larger datasets while giving significantly faster convergence as compared to the existing algorithms. Since, extremal points have the sparsest convex representation, archetypes are identified by projecting the data in a linearly transformed/coefficient space involving sparse matrices. Here, appropriate sparse exemplars are selected by employing an iterative fast subset selection approach.**

I. Introduction

Archetypal analysis (AA) is decomposition of data as convex combinations of extremal points/archetypes $\mathbf{d}_j$, which lie on the convex hull of the data and are themselves restricted to being a convex combinations of individual observations $\mathbf{x}_i$ [1] by solving the following optimization problem with simplex constraints:

$$\underset{\substack{\mathbf{B},\mathbf{A}\\ \mathbf{b}_j\in\Delta_l,\mathbf{a}_i\in\Delta_d}}{\operatorname{argmin}} \|\mathbf{X}-\mathbf{DA}\|_F^2 = \|\mathbf{X}-\mathbf{XBA}\|_F^2 = \sum_i^l \|\mathbf{x}_i - \mathbf{D}\mathbf{a}_i\|_2^2,$$
$$\Delta_l \triangleq [\mathbf{b}\succeq 0, \|\mathbf{b}\|_1 = 1], \Delta_d \triangleq [\mathbf{a}\succeq 0, \|\mathbf{a}\|_1 = 1] \tag{1}$$

Here, the columns of $\mathbf{D}$ are the inferred archetypes. This problem can be solved iteratively using quadratic programming (QP) in an alternating minimization framework [2], [3], [1]. In contrast to using only generic QP solvers or gradient descent based algorithms, we leverage the underlying *sparseness* property of the solution, to design a greedy algorithm having significantly faster convergence and scalablity to larger datasets.

The proposed GAA algorithm is computationally efficient because it employs coefficient space learning [4], [5], which involve solving the following optimization problem only once

$$\begin{aligned}&\underset{\mathbf{c}_i\in\Delta_l}{\operatorname{argmin}} \|\mathbf{X}-\mathbf{XBA}\|_F^2 = \|\mathbf{X}-\mathbf{XC}\|_F^2 = \\ &\|vec(\mathbf{X}) - (\mathbf{I}\otimes\mathbf{X})vec(\mathbf{C})\|_2^2 \text{ s.t. } diag(\mathbf{C}) = 0,\end{aligned} \tag{2}$$

Here, $vec(.)$ denotes the vectorization operation, and matrix $\mathbf{C}$ can be seen as the coefficient matrix representing each exemplar as a linear combination of others [4]. The columns $\mathbf{c}_i$s of the coefficient matrix $\mathbf{C}$ are computed such that the error is bounded i.e., $\|\mathbf{X}-\mathbf{XC}\|_F^2 < \eta$[1], and the factor $\mathbf{B}$ is updated by solving the following problem for a fixed $\mathbf{A}$

$$\underset{\mathbf{B},\mathbf{b}_j\in\Delta_l}{\operatorname{argmin}} \|\mathbf{C}-\mathbf{BA}\|_F^2, \tag{3}$$

Hence, updating $\mathbf{B}$ is independent of any computations involving $\mathbf{X}$. Similarly, for a fixed $\mathbf{B}$, $\mathbf{A}$ can be updated via a suitable fast QP solver[2]. Further, note that all the involved matrices i.e $\mathbf{C}$, $\mathbf{B}$ and $\mathbf{A}$ are sparse or compressible, which helps in speeding up the algorithm.

[1]There exist a case where $\mathbf{XC} = \mathbf{XBA}$, but $\mathbf{C} \neq \mathbf{BA}$. This occurs when $\mathbf{C} = \mathbf{BA} + \mathbf{V}$, where $\mathbf{V}$ lies in the null space of $\mathbf{X}$.

[2]GAA employ the active-set QP solver: http://spams-devel.gforge.inria.fr/

A. Finding Archetypes using Subset Selection

GAA identifies archetypes on convex hull by exploiting the intrinsic sparsity structure of convex representations. It is based on the fact that *extremal points have a sparser convex representation as compared to interior points of the data distribution*. The geometric interpretation of our approach is depicted in Figure 1. Hence, $\mathbf{B}$ is updated (in the coefficient domain) column wise by sequentially extracting a new sparse column $\mathbf{e}_k$ from the current error matrix $\mathbf{E}^\Omega$ (Step 4 in Algorithm 1). Here, set $\Omega = |\mathcal{S}(\mathbf{a}_{[j]})|$, ($\mathcal{S}$ being the soft-thresholding operator) leverages the underlying sparsity pattern in row $\mathbf{a}_{[j]}$ of $\mathbf{A}$ and favors the observations closer to edges, leading to a better estimate of archetypes. Note that after each selection coefficients $\mathbf{a}_{[j]}^\Omega$ are not re-estimated, as the goal is just to emphasize the potential candidates for next atom update. The GAA algorithm can also be extended to robust and relaxed AA model [1], [3].

II. Experimental Results

This section compares the efficiency of the proposed GAA algorithm along with existing algorithms i.e., AA using active-set (AAAS) algorithm [2], AA using projected gradient (AAPG) algorithm [1], AA with Kullback-Leibler divergence (AAKL) algorithm [6], in various signal processing/machine learning applications[3].

Experiment1: Fig. 2 and 3 shows the AA results on synthetic and real dataset.

Experiment2: Table I shows a comparison of CPU run-times on two real datasets[4]. In practice, it was observed that the empirical complexity of GAA algorithm is linear in l and d, while for AAAS algorithm it is only linear in l.

Experimental results confirms that GAA performs comparable to existing methods with significant time complexity gain.

Algorithm 1 Greedy Archetypal Analysis (GAA) algorithm

Inputs: Training signal matrix $\mathbf{X}\in\mathbb{R}^{n\times l}$
Outputs: $\mathbf{D}\in\mathbb{R}^{n\times d}$, $\mathbf{B}\in\mathbb{R}^{l\times d}$ and $\mathbf{A}\in\mathbb{R}^{d\times l}$
Initialization: η, $iter$, $\mathbf{D}$, random $\mathbf{B}$ s.t. $\mathbf{D} = \mathbf{XB}$ and $\mathbf{C}$ via (2).

Perform outer iterations
1: $\mathbf{A} \leftarrow \underset{\mathbf{A},\mathbf{a}_i\in\Delta_d}{\operatorname{argmin}} \|\mathbf{X}-\mathbf{DA}\|_F^2$, $\mathbf{E}\leftarrow\mathbf{C}$, $\mathcal{I} = \emptyset$
Perform inner iterations: $j = 1$
2: $\Omega \leftarrow |\mathcal{S}(\mathbf{a}_{[j]})|$, $k \leftarrow \text{argmax}(Gini(\mathbf{e}_k))$ $k\notin\mathcal{I}, k\in\Omega$
3: $\mathbf{b}_j \leftarrow \mathbf{e}_k, \mathbf{b}_j \leftarrow \mathbf{b}_j/\|\mathbf{b}_j\|_2, \mathbf{E}^\Omega \leftarrow \mathbf{E}^\Omega - \mathbf{b}_j\mathbf{a}_{[j]}^\Omega$
4: $\mathcal{I}\leftarrow\mathcal{I}\cup k, j = j+1$
Until d **columns**
5: $\mathbf{D}\leftarrow\mathbf{XB}$, $iter \leftarrow iter - 1$
Until $iter > 0$

[3]Comparison done on a Quad-Core Intel i7 machine at 3.5 GHz, 12 GB RAM, using MATLAB and under Win10 operating system.

[4]http://statweb.stanford.edu/ tibs/ElemStatLearn/data.html
https://cs.brown.edu/ gen/sunattributes.html

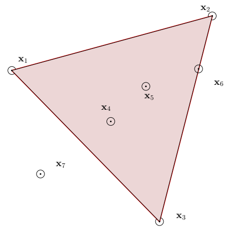
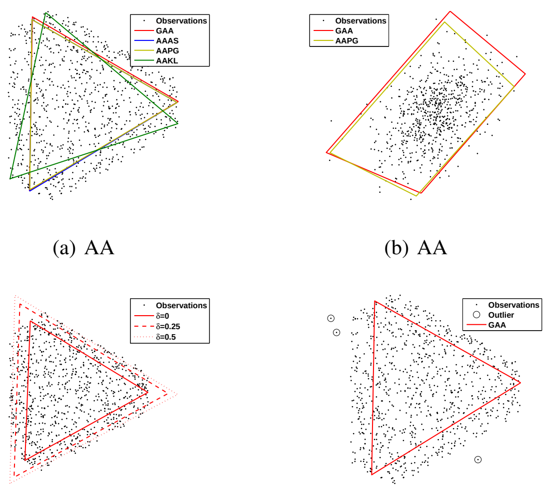


Fig. 1. Illustration of geometry in convex representation using a 2-simplex and 7 points in a 2-D plane. Convex hull is marked by red boundary. Points $\mathbf{x}_4$ and $\mathbf{x}_5$ can be represented as a convex combination of points $\mathbf{x}_1, \mathbf{x}_2$ and $\mathbf{x}_3$, while point $\mathbf{x}_6$ as a convex combination of $\mathbf{x}_2$ and $\mathbf{x}_3$. Point $\mathbf{x}_7$ can only be represented as an affine combination of other points.



(a) AA

(b) AA

(c) Relaxed-AA [1] using GAA

(d) Robust-AA [3] using GAA

Fig. 2. Illustration of archetypal analysis for real-valued 1000 observations where model order 'd' in (a), (c), (d) is 3, and in (b) is 4. The corners of each colored polygon indicate the estimated archetypes. Note that the relaxed AA works best for the model order-3 dataset which has no true archetypes.

References

[1] Morten Mrup and Lars Kai Hansen, "Archetypal analysis for machine learning and data mining," *Neurocomputing*, vol. 80, pp. 54 – 63, 2012, Special Issue on Machine Learning for Signal Processing 2010. 1, 2

[2] Yuansi Chen, Julien Mairal, and Zaid Harchaoui, "Fast and robust archetypal analysis for representation learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Washington, DC, USA, June 2014, CVPR '14, pp. 1478–1485, IEEE Computer Society. 1

[3] Manuel J. A. Eugster and Friedrich Leisch, "Weighted and robust archetypal analysis," *Computational Statistics and Data Analysis*, vol. 55, no. 3, pp. 1215 – 1225, 2011. 1, 2

[4] V. Abrol, P. Sharma, and A. K. Sao, "Greedy dictionary learning for kernel sparse representation based classifier," *Pattern Recognition Letters*, vol. 78, pp. 64 – 69, 2016. 1

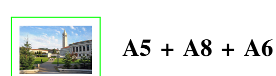
[5] V. Abrol, P. Sharma, and A. K. Sao, "Greedy double sparse dictionary learning for sparse representation of speech signals," *Speech Communication*, vol. 85, pp. 71 – 82, December 2016. 1

[6] A. Diment and T. Virtanen, "Archetypal analysis for audio dictionary learning," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, October 2015, pp. 1–5. 1

TABLE I
AVERAGE ARCHETYPAL ANALYSIS RUN-TIMES FOR FINDING 1000 ARCHETYPES VIA DIFFERENT METHODS OVER 10 TRIALS.

Dataset	**Number of Observations**	**Run-time (s)**		
		AAPG	AAAS	GAA
USPS	9298	1500	950	520
SUN Attribute	14340	2250	1400	820

(a)



(b)

Fig. 3. Visualization of the archetypes found for the SUN attribute dataset. (a) The top six generating images for each one of the ten archetypes (A1-A10). (b) Example image with top three generating archetypes.