A Geometric Approach to Archetypal Analysis via Sparse Projections

V. Abrol [1] **P. Sharma** [2]

Abstract

Archetypal analysis (AA) aims to extract patterns using self-expressive decomposition of data as convex combinations of extremal points (on the convex hull) of the data. This work presents a computationally efficient greedy AA (GAA) algorithm. GAA leverages the underlying geometry of AA, is scalable to larger datasets, and has significantly faster convergence rate. To achieve this, archetypes are learned via sparse projection of data. In the transformed space, GAA employs an iterative subset selection approach to identify archetypes based on the sparsity of convex representations. The work further presents the use of GAA algorithm for extended AA models such as robust and kernel AA. Experimental results show that GAA is considerably faster while performing comparable to existing methods for tasks such as classification, data visualization/categorization.

1. Introduction

In recent years, various matrix decompositions techniques have helped researchers in summarizing and visualizing large datasets of natural scenes, objects, faces, videos, and text (Elhamifar et al., 2012; Mørup & Hansen, 2012; Thurau & Bauckhage, 2009). The popular approaches are clustering methods, principal component analysis (PCA), independent component analysis (ICA), dictionary learning (DL)/sparse coding (SC), non-negative matrix factorization (NMF) etc (Bernstein, 2009; Mørup & Hansen, 2012; Tosic & Frossard, 2011). Specifically, we seek factors $\mathbf{D} \in \mathbb{R}^{n\times d}$ and $\mathbf{A} \in \mathbb{R}^{d\times l}$ for a collection of signals as columns of matrix $\mathbf{X} \in \mathbb{R}^{n\times l}$ by minimizing the objective function

$$\|\mathbf{X}-\mathbf{D}\mathbf{A}\|_F^2 = \sum_{i=1}^{l} \|\mathbf{x}_i - \mathbf{D}\mathbf{a}_i\|_2^2, \qquad (1)$$

where $\|.\|_F$ is the Frobenius norm (Tosic & Frossard, 2011; Elhamifar et al., 2012). Different decomposition approaches employ different constraints (such as non-negativity, sparsity, independence etc.) on factors $\mathbf{D}$ and $\mathbf{A}$ and hence lead to different type of representations of the data, suitable for various pattern recognition tasks. For instance, clustering approaches give easy interpretable representation, while approaches such as PCA/ICA/NMF/SC are more efficient in capturing inherent structures and patterns of data (Mørup & Hansen, 2012).

This paper is focused on an unsupervised learning technique called archetypal analysis (AA), which is intuitive and easy to interpret like clustering, and has flexibility as that of matrix factorization (Chen et al., 2014; Mørup & Hansen, 2012; Fotiadou et al., 2017). In contrast to centroids, archetypes characterize extremal rather than average properties of the given data, and therefore leads to a more compact representation (Seth & Eugster, 2016a; Yale Song et al., 2015). AA is decomposition of data as convex combinations of extremal points that lie on the convex hull of the data and are themselves restricted to being a convex combinations of individual observations (Mørup & Hansen, 2012; Mei et al., 2018). AA has found application in variety of problems ranging from style transfer (Wynen et al., 2018), hyperspectral image unmixing (Zhao et al., 2016; Zhao et al., 2016), fMRI analysis (Hinrich et al., 2016), video summarization (Yale Song et al., 2015), clustering (Mørup & Hansen, 2012; Seth & Eugster, 2016a), acoustic modelling (Thakur et al., 2018) to action and texture segmentation (Cabero & Epifanio, 2019; Fotiadou et al., 2017). However, compared to other factorization models, it is believed that the lack of efficient algorithms has limited the deployment of AA to prevail as a tool for data visualization and analysis (Chen et al., 2014); our goal in this paper is to address this issue.

This paper presents an approach which exploits the underlying geometry and sparsity pattern of the convex representations to identify archetypes or points on convex hull of the data. It is based on the observation that extremal points have a sparser convex representation compared to interior points of the data distribution. This motivates identifying archetypes efficiently in the transformed space involving sparse matrices. First, we develop an efficient algorithm based on greedy column subset selection strategy (Jafari & Plumbley, 2011). Then, we demonstrate that our approach

[1] Mathematical Institute, University of Oxford. [2] Department of Engineering Science, University of Oxford. Correspondence to: <abrol@maths.ox.ac.uk>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

is scalable and faster than existing publicly available AA algorithms. Finally, we show the application of AA in various problems ranging from computer vision, digit classification to data visualization.

2. Problem Formulation

The AA problem attempts to identify the archetypes as columns of matrix $\mathbf{D}$, i.e., a factorial representation to the data matrix $\mathbf{X}$ under two geometrical constraints: 1) each observation vector $\mathbf{x}_i$ should be well approximated by a convex combination of archetypes $\mathbf{d}_j$s, and 2) each archetype $\mathbf{d}_j$ should be a convex combination of observations $\mathbf{x}_i$s (Chen et al., 2014; Eugster & Leisch, 2011; Mørup & Hansen, 2012). To elaborate, AA is equivalent to solving the following non-convex optimization problem with simplex constraints:

$$\begin{aligned} &\underset{\substack{\mathbf{B},\mathbf{A} \\ \mathbf{b}_j\in\Delta_l,\mathbf{a}_i\in\Delta_d}}{\operatorname{argmin}} \quad \|\mathbf{X}-\mathbf{D}\mathbf{A}\|_F^2 = \|\mathbf{X}-\mathbf{X}\mathbf{B}\mathbf{A}\|_F^2, \\ &\Delta_l \triangleq [\mathbf{b}\succeq 0, \|\mathbf{b}\|_1 = 1], \Delta_d \triangleq [\mathbf{a}\succeq 0, \|\mathbf{a}\|_1 = 1] \end{aligned} \tag{2}$$

Here, the columns of $\mathbf{D}$ are the inferred archetypes. For $d = 1$, the solution is given by centroid of the data, for $d = 2$, the solution coincides with the first principal axis of the data, and for $d > 2$, archetypes lie on the convex hull of the observations (Chen et al., 2014; McCallum & Avis, 1979). Further, the stochastic constraints in (2) enforces sparseness i.e., only a few of the observations in $\mathbf{X}$ will contribute to $\mathbf{d}_j$, and similarly columns $\mathbf{a}_i$ of matrix $\mathbf{A}$ are convex and sparse. This problem can be solved using quadratic programming (QP). This is done via alternate minimization, as the problem is convex with respect to one of the variables $\mathbf{B}$ or $\mathbf{A}$, when the other is fixed.

2.1. Prior Work

AA dates back to alternating least-squares based algorithm by Cutler and Breiman (Adele Cutler, 1994). In the past there have been numerous improvements; specifically AA using projected gradients (AAPG) (Mørup & Hansen, 2012), with Kullback-Leibler divergence (AAKL) (Diment & Virtanen, 2015), efficient active-set quadratic programming (AAAS) (Chen et al., 2014), projection-free convex optimization via Frank-Wolfe techniques (Bauckhage et al., 2015) and online dictionary learning via block-coordinate descent (AAODL) (Mei et al., 2018). However, these approaches do not provide theoretical guarantees on the quality of approximation. Some approaches trade-off accuracy for speed to provide approximate solutions by performing AA on precomputed data subset e.g., Frame (Mair et al., 2017) or Coreset (Mair & Brefeld, 2019). Again for large problem size e.g., the Million Song Dataset, pre-computing Frame (all data points lying on the boundary of the convex hull) is difficult within a reasonable amount of time while AA Coreset results in large approximation error (Mair & Brefeld, 2019).

Notably there also exists other versions of AA such as probabilistic AA (Seth & Eugster, 2016b), functional AA (Moliner & Epifanio, 2019), separable AA (Damle & Sun, 2017) or AA for missing data (Epifanio et al., 2019), which are not considered in this work.

3. Proposed Greedy Archetypal Analysis (GAA) Algorithm

AA is a least-squares optimization problem with simplex constraints. In contrast to using only generic QP solvers or gradient decent based algorithms, we leverage the underlying sparsity of convex representations, to design an algorithm which is scalable to larger datasets while giving significantly faster convergence as compared to existing methods. In addition, the usual approach to solve (2) is to optimize both factors $\mathbf{B}$ and $\mathbf{A}$ alternatively with respect to the whole data matrix $\mathbf{X}$. Thus, computational complexity of conventional AA algorithms scales exponentially with the dimensionality of $\mathbf{X}$ (McCallum & Avis, 1979).

We show that alternative to (2), one can define an objective function such that learning $\mathbf{B}$ is efficient and independent of any computations involving $\mathbf{X}$. Once $\mathbf{B}$ is known, $\mathbf{A}$ can be updated via a suitable fast QP solver[1]. This is achieved by learning the matrix $\mathbf{B}$ in the coefficient space rather than the signal space (Abrol et al., 2016). This is motivated from the fact that the error in the coefficient or the representation domain upper bounds the error in the signal domain. In particular for a matrix $\mathbf{D}$ we have:

$$\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2 = \|\mathbf{D}\mathbf{a}-\mathbf{D}\hat{\mathbf{a}}\|_2^2 \leq \lambda^2(\mathbf{D})\|\mathbf{a}-\hat{\mathbf{a}}\|_2^2 \tag{3}$$

where $\hat{\mathbf{x}}$ is an estimate of $\mathbf{x}$ and $\lambda(\mathbf{D})$ is the largest singular value of $\mathbf{D}$ (Chen et al., 2013). A similar approach using random projections has been shown to be effective in case of NMF (Chu & Lin, 2008; Thurau et al., 2011). Now consider an alternate objective function by re-expressing (2) as:

$$\begin{aligned} &\|\mathbf{X}-\mathbf{X}\mathbf{B}\mathbf{A}\|_F^2 = \|\mathbf{X}-\mathbf{X}\mathbf{C}\|_F^2 \\ &\text{s.t. } diag(\mathbf{C}) = 0, \ \mathbf{c}_i \succeq 0, \text{ and } \|\mathbf{c}_i\|_1 = 1 \end{aligned} \tag{4}$$

Here, $diag(.)$ denotes the diagonal elements and matrix $\mathbf{C}$ (having columns $\mathbf{c}_i$), can be seen as the coefficient matrix for representing each exemplar in $\mathbf{X}$ as a linear combination of other exemplars (Abrol et al., 2016). This can be interpreted as an affinity transformation, where training exemplars lying in the same subspace utilize one another in their convex representations (Elhamifar et al., 2012). The coefficient matrix $\mathbf{C}$ is computed such that the error is bounded i.e.,

[1] GAA employ the active-set solver for QP from SPAMS toolbox: http://spams-devel.gforge.inria.fr/

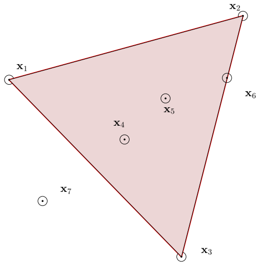


Figure 1. Illustration of geometry in convex representation using a 2-simplex and 7 points in a 2-D plane.

$\|\mathbf{X} - \mathbf{XC}\|_F^2 < \eta$ [2]. To maintain this bound, the product $\mathbf{BA}$ should be close to $\mathbf{C}$ for a convex coefficient matrix $\mathbf{A}$, or equivalently we can update the factors $\mathbf{B}$ and $\mathbf{A}$ by alternatively solving the problem (Abrol et al., 2016)

$$\underset{\substack{\mathbf{B},\mathbf{A} \\ \mathbf{b}_j \in \Delta_l, \mathbf{a}_i \in \Delta_d}}{\text{argmin}} \quad \|\mathbf{C} - \mathbf{BA}\|_F^2, \tag{5}$$

with respect to $\mathbf{B}$ and $\mathbf{A}$, instead of the one defined in (2). Here, matrix $\mathbf{C}$ is computed using $\mathbf{X}$ only once. Interestingly, computing $\mathbf{C}$ gives an inherent advantage of clustering the data by applying spectral clustering to the graph Laplacian of $\mathbf{G} = |\mathbf{C}| + |\mathbf{C}^T|$. Hence, if partially labelled data is available, one can know the suitability of the obtained $\mathbf{C}$ prior to solving (5). There exist a case where $\mathbf{XC} = \mathbf{XBA}$, but $\mathbf{C} \neq \mathbf{BA}$, which occurs when $\mathbf{C} = \mathbf{BA} + \mathbf{V}$ with $\mathbf{V} \in Null(\mathbf{X})$. The problem in (5) is an alternate formulation of (2), and can also be seen as a matrix factorization problem. The main difference lies in use of $\mathbf{C}$ instead of $\mathbf{X}$. Further, note that all the matrices involved i.e $\mathbf{C}$, $\mathbf{B}$ and $\mathbf{A}$ are sparse or compressible which helps in speeding up the procedure of finding archetypes as discussed in the next section 3.1.

3.1. Finding Archetypes using Subset Selection

We employ a greedy subset/exemplar selection approach to update $\mathbf{B}$ i.e., the training exemplars (or equivalently columns of $\mathbf{C}$) are chosen as columns/atoms of $\mathbf{B}$. Further, it is ensured that the information learned by the previous columns is used to guide an adaptive selection of subsequent ones. The aim is to identify archetypes or points on convex hull by exploiting the intrinsic sparsity structure of convex representations. The assumption of sparsity comes from the observation that archetypes are the extremal points of the data, and hence only a few archetypes are sufficient to represent an observation (Seth & Eugster, 2016a). To achieve this, define the overall error/residual matrix in the coefficient domain as,

$$\mathbf{E} = \mathbf{C} - \mathbf{BA} = \mathbf{C} - (\mathbf{W_1} + \mathbf{W_2} + \ldots) \; \forall_j : \mathbf{W}_j = \mathbf{b}_j \mathbf{a}_{[j]}, \tag{6}$$

where, $\mathbf{b}_j$ and $\mathbf{a}_{[j]}$ denotes the j^{th} column and row of $\mathbf{B}$ and $\mathbf{A}$, respectively.

In each iteration, matrix $\mathbf{B}$ and $\mathbf{A}$ are alternatively optimized to minimize the residual. In general, careful initialization improves the convergence rate and reduces the risk of finding inappropriate archetypes. Hence, as suggested in (Mørup & Hansen, 2012), we used the 'FurthestSum' method to initialize $\mathbf{D}$. Initially, the coefficient matrix $\mathbf{A}$ is updated by a fast QP solver. Following this, the error/residual $\mathbf{E}$ is initialized to $\mathbf{C}$ i.e., considering $\mathbf{B} = \emptyset$. Finally, $\mathbf{B}$ is updated column-by-column by sequentially extracting a new column $\mathbf{e}_k$ from the current error matrix $\mathbf{E}^\Omega$ based on the criterion; maximum $Gini(\mathbf{e}_k)$, computed over its columns[3]. Since, $\mathbf{C}$ is sparse in nature, the sparsity measure is used to quantify information content of its columns in order to update $\mathbf{B}$.

The geometric interpretation of our selection criterion can be understood from Figure 1, which shows a 2-simplex and 7 points in a 2-D plane (4-on, 2-inside and 1-outside of convex hull marked by red boundary). Following properties of convex geometry, 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 $\mathbf{x}_3$. In fact, every convex combination of two extremal points lies on the line segment between the points. However, point $\mathbf{x}_7$ can only be represented as an affine combination of all other points. It is now evident that extremal points have a sparser convex representation compared to interior points of the data distribution, which motivates our selection criterion. Further, note that any finite set of non-negative vectors (columns of $\mathbf{C}$) lies within a convex polyhedral cone, and thus columns of $\mathbf{B}$ are indeed edges of the cone that coincide with them.

Further, to leverage the underlying sparsity pattern in $\mathbf{a}_{[j]}$, any column $\mathbf{b}_j$ is updated from only those columns in $\mathbf{E}$ (denoted by set $\Omega = |\mathcal{S}(\mathbf{a}_{[j]})|$, $\mathcal{S}$ being the soft-thresholding operator), whose representations use the current archetype. Thresholding favours the observations closer to edges as they are sufficient to identify the extremal points of the data distribution, leading to a better estimate of archetypes. Following this, the error $\mathbf{E}$ is minimized by subtracting the selected column's energy contribution i.e., $\mathbf{E}^\Omega_{new} = \mathbf{E}^\Omega_{old} - \mathbf{b}_j \mathbf{a}^\Omega_{[j]}$. Note that the coefficients $\mathbf{a}^\Omega_{[j]}$ are not re-estimated, as the main goal of the GAA algorithm is just to emphasize the potential candidates for next atom update. This is done to give more emphasis to the current updated support set, such that the chosen archetypes in addition to

[2] (4) can also be formulated as a standard QP problem.

[3] Function $Gini(.)$ denotes the Gini Index sparsity metric which has been shown to satisfy all important sparsity attributes (Hurley & Rickard, 2009).

Algorithm 1 Greedy Archetypal Analysis (GAA) algorithm

Inputs: Training signal matrix $\mathbf{X} \in \mathbb{R}^{n\times l}$
Outputs: Archetypal dictionary $\mathbf{D} \in \mathbb{R}^{n\times d}$ and sparse coefficient matrices $\mathbf{B} \in \mathbb{R}^{l\times d}$ and $\mathbf{A} \in \mathbb{R}^{d\times l}$
Initialization: δ, $iter$, $\mathbf{D}$ via FurthestSum, random $\mathbf{B}$ s.t. $\mathbf{D} = \mathbf{XB}$

```
Compute $\mathbf{C}$ via (4)
Perform outer iterations
    $\mathbf{A} \leftarrow \underset{\mathbf{A},\mathbf{a}_i\in\Delta_d}{\operatorname{argmin}} \|\mathbf{X}-\mathbf{DA}\|_F^2$
    $\mathbf{E} \leftarrow \mathbf{C}, \mathcal{I} = \emptyset$
    Perform inner iterations: $j = 1$
        $\Omega \leftarrow |\mathcal{S}(\mathbf{a}_{[j]})|$
        $k \leftarrow \underset{k\notin\mathcal{I},k\in\Omega}{\operatorname{argmax}}(Gini(\mathbf{e}_k))$
        $\mathbf{b}_j \leftarrow \mathbf{e}_k, \mathcal{I} \leftarrow \mathcal{I} \cup k$
        $\mathbf{b}_j \leftarrow \mathbf{b}_j/\|\mathbf{b}_j\|_1$
        $\mathbf{E}^{\Omega} \leftarrow \mathbf{E}^{\Omega} - \mathbf{b}_j\mathbf{a}_{[j]}^{\Omega}$
        $j = j + 1$
    Until $d$ columns
Until $iter > 0$
$\mathbf{D} \leftarrow \mathbf{XB}$
$\mathbf{A} \leftarrow \underset{\mathbf{A},\mathbf{a}_i\in\Delta_d}{\operatorname{argmin}} \|\mathbf{X}-\mathbf{DA}\|_F^2$
```

being extremal, are encouraged to be close to the data points that uses them in their decompositions. The modified error matrix serves for the next atom update and as a result, the same exemplar will not be selected again. The pseudo-code of the proposed approach is shown in Algorithm 1.

4. Solution to Extended AA models

In many scenarios, it is not possible to find the "true" archetypes or a convex representation in terms of the observed data (Mørup & Hansen, 2012). Also, conventional AA is not robust to outliers, and may produce undesirable archetypes. Furthermore, finding archetypes sometimes is much easier in higher dimensions using a measure of pairwise similarity (Chen et al., 2014). To address these issues, various extensions of the proposed approach for extended AA models are briefly discussed, however a full evaluation of these models is beyond the scope of this manuscript.

4.1. Relaxed AA Model

To address the issue of non-existence of true archetypes, work in (Mørup & Hansen, 2012) proposed a relaxed AA model. This model assumes that the true archetypes reside outside the convex hull of the data. Mathematically, the model can be expressed as:

$$\begin{aligned} &\underset{\substack{\mathbf{B},\mathbf{A}\\ \mathbf{b}_j\in\Delta_l,\mathbf{a}_i\in\Delta_d}}{\operatorname{argmin}} \|\mathbf{X}-\mathbf{DA}\|_F^2 = \|\mathbf{X}-\mathbf{XBUA}\|_F^2, \\ &\qquad \boldsymbol{\alpha} = diag(\mathbf{U}),\ \ 1-\delta \le \alpha_j \le 1+\delta, \end{aligned} \tag{7}$$

where α_j is the scaling parameter which relaxes $\|\mathbf{b}_j\|_1$ within the range defined by δ (see Figure 2(c) for illustration). The solution to (7) can be found using the same method as proposed in Section 3, where in each iteration after estimating $\mathbf{B}$ and $\mathbf{A}$, one can update $\boldsymbol{\alpha}$ via gradient descend (Mørup & Hansen, 2012).

4.2. Robust AA Model

In the conventional AA model, one tries to minimize the Euclidean (matrix) norm of the residual. Hence, the AA algorithm will be biased towards outliers, and ends up finding undesired archetypes (Eugster & Leisch, 2011). This can be addressed by replacing the least-square function by another function that reduces the effect of outliers. One such robust AA model was proposed in (Chen et al., 2014), and is expressed as:

$$\begin{aligned} \underset{\substack{\mathbf{B},\mathbf{A}\\ \mathbf{b}_j\in\Delta_l,\mathbf{a}_i\in\Delta_d}}{\operatorname{argmin}} &= \sum_i h(\|\mathbf{x}_i - \mathbf{D}\mathbf{a}_i\|_2) \\ &= .5\sum_i \frac{1}{w_i}\|\mathbf{x}_i - \mathbf{D}\mathbf{a}_i\|_2^2 + w_i, \end{aligned} \tag{8}$$

where $h(u) = .5\min_{w\ge\epsilon}[u^2/w + w] : \mathbb{R} \to \mathbb{R}$ is the Huber loss function. With definition of (8) in place, it is easy to see that for fixed $\mathbf{B}$ and $\mathbf{A}$, we have a closed form solution for weights w_i as $\max(\|\mathbf{x}_i - \mathbf{D}\mathbf{a}_i\|_2^2, \epsilon)$ (Chen et al., 2014). Similarly, while fixing w_i, $\mathbf{B}$ and $\mathbf{A}$ can be alternatively optimized as proposed in Section 3.

4.3. Kernel AA Model

The conventional AA model can also be generalized to the case when finding archetypes is much easier in some higher dimensional Hilbert space (Bauckhage & Manshaei, 2014). For instance, data lying in overlapping union of subspaces can be projected in higher dimensional space to separate the individual subspaces in order to extract meaningful archetypes for tasks such as visualization and classification (Abrol et al., 2016). To understand this, consider the objective function in (2) as:

$$\begin{aligned} &\|\phi(\mathbf{X}) - \phi(\mathbf{X})\mathbf{BA}\|_F^2 = \|\phi(\mathbf{X})(\mathbf{I}-\mathbf{BA})\|_F^2 \\ &=\mathbf{tr}((\mathbf{I}-\mathbf{BA})^T\mathcal{K}(\mathbf{X},\mathbf{X})(\mathbf{I}-\mathbf{BA})) \end{aligned} \tag{9}$$

where the transformation $\phi : \mathbb{R}^n \to \mathcal{R}$ maps the input space to a high-dimensional Hilbert space $\mathcal{R}$ (Abrol et al., 2016; Van Nguyen et al., 2013). Although the transformation ϕ

Table 1. Average archetypal analysis run-times for finding 1000 archetypes via different methods over 10 trials.

Dataset	**Samples**	**Run-time (s)**				
		AAKL	AAPG	AAAS	AAODL	GAA
SUN Attribute	14340	2400	2250	1400	1100	820
Flickr	70K	8500	8030	5740	3710	3230
H3.6M	300K	67452	53130	34040	21340	17050

is unknown, the optimization of kernel AA model is still feasible via kernel trick using the kernel matrix $\mathcal{K}(\mathbf{X}, \mathbf{X})$ whose elements are computed as $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$, κ being the kernel similarity function (Van Nguyen et al., 2013; Zhao et al., 2016). Again the objective in (9) can be minimized and the solution can be obtained in coefficient domain using the method proposed in Section 3.

5. Computational Complexity

Projected gradient based AAPG algorithm update both the factors $\mathbf{B}$ and $\mathbf{A}$ as a whole. The mathematical complexity of AAPG algorithm thus scales as $\mathcal{O}(ndl)$. In contrast, factors $\mathbf{B}$ and $\mathbf{A}$ are updated column by column in both GAA and AAAS algorithms. In AAAS algorithm which essentially is a cyclic-coordinate algorithm, both the factors are updated using a fast active-set based QP solver. For instance, the complexity of solving for a column of $\mathbf{A}$ approximately scales as $\mathcal{O}(nd + a^2)$, a being the size of the active set in the current iteration (see (Chen et al., 2014) for more details). The difference with GAA lies in the update of factor $\mathbf{B}$. In GAA algorithm this is done by firstly sorting the vector in $\mathcal{O}(n \log n)$ operations, precomputing the Gini Index measure for each observation with complexity of $\mathcal{O}(n)$. Next, the archetypes are identified by subset selection (finding the vector with minimum sparsity) with complexity $\mathcal{O}(|\Omega \setminus \mathcal{I}|)$.

Table 1 shows runtime comparison of different algorithms for a fixed error tolerance of 10^{-3} as stopping criterion. The empirical computational times are measured on a Quad-Core Intel i7 machine at 3.5 GHz, 12 GB RAM, using MATLAB and under Windows10 operating system. Experimental results shows that GAA is nearly $2\times$ faster than the current state-of-the-art AAAS algorithm, while on large datasets[4] it is considerably faster than recently proposed AAODL algorithm. In practice, it was observed that the empirical complexity of GAA algorithm is linear in l and d, while for AAAS and AAODL algorithm it is only linear in l. As discussed later in Section7, matrix $\mathbf{C}$ can be precomputed over data partitions in parallel on multiple machines, resulting in further speed gains.

6. Convergence Analysis

GAA employ active-set based QP algorithm for computing factors $\mathbf{C}$ and $\mathbf{A}$, which is guaranteed to converge to a stationary point. In regard to $\mathbf{B}$, exploiting the approximate sub-modularity property of the matrix factorization (Nemhauser et al., 1978; Krause et al., 2008), approximation guarantee on how the proposed algorithm for updating $\mathbf{B}$ will behave in practice can be obtained.

Proposition1: *Let's denote the candidate training set by $\mathcal{T}$, the selected and optimal archetypal set by $\mathcal{A}$ and $\mathcal{A}^*$, respectively. GAA algorithm while minimizing (2) starts with an empty set $\mathcal{A} = \emptyset$, iteratively adds a new element and obtains a set $\mathcal{A}$ which is at-least a constant fraction of the optimal one.*

Proposition1 holds for sub-modular functions and it is easy to show that (2) has a sub-modular surrogate function which is also related to the incoherence, a geometric property of the candidate training set (Nemhauser et al., 1978). To investigate how good this theoretical approximation is, an experiment is done to check the ability of different algorithms to recover the true underlying archetypal dictionary. A data matrix $\mathbf{X} \in \mathbb{R}^{200\times 30000}$ is generated, by selecting uniformly random 5 archetypes from $\mathbf{D} \in \mathbb{R}^{200\times 100}$. By solving the inverse problem, we were able to recover the true $\mathbf{D}$ more than 74.7% (AAODL) (Mei et al., 2018), 75% (AAAS) (Chen et al., 2014), 74.4% (AAPG) (Mørup & Hansen, 2012), and 72.9% (GAA) of the time.

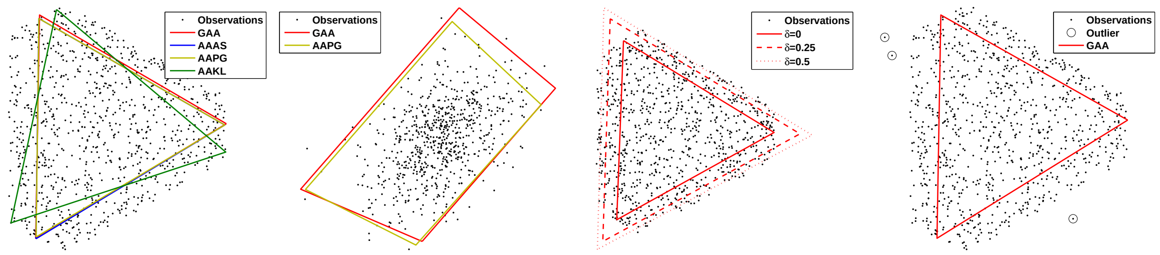
Proposition2: *GAA algorithm restricts its search of archetypes around the convex hull of the data.*

Result in Proposition2 is interesting in the sense that GAA searches for archetypes around the convex hull regardless of the distribution of data points inside the simplex and hence is scalable for larger datasets. This suggest various alternatives to further speed up the archetypal search, few of which are discussed in the next section.

7. Sequential Updates for GAA

In the proposed method one needs to solve for $\mathbf{C}$, and it may be argued if solving for a big matrix $\mathbf{C}$ is more favorable than solving for the smaller matrices $\mathbf{A}$ and $\mathbf{B}$ or not. How-

[4] The flickr dataset was obtained by querying "most interesting" images across 50 categories.
H3.6M Dataset: http://vision.imar.ro/human3.6m/
SUN Attribute: https://cs.brown.edu/$\sim$gmpatter/sunattributes.html



(a) AA (b) AA (c) Relaxed-AA using GAA (d) Robust-AA using GAA

Figure 2. Illustration of archetypal analysis for real-valued observations where model order in (a), (c), (d) is 3, and in (b) is 4. The corners of each coloured polygon indicate the estimated archetypes.

ever, note that in existing algorithms such as in (Chen et al., 2014), updating $\mathbf{B}$ require operations involving $\mathbf{X}^T$ and $(\mathbf{X}^T\mathbf{X})^{-1}$ which has large time and storage complexity for larger datasets. Moreover, such operations are performed iteratively until both factors $\mathbf{A}$ and $\mathbf{B}$ converges. In contrast, in GAA algorithm $\mathbf{C}$ is computed only once, which takes away the computational burden from step involving update of $\mathbf{B}$, done via fast subset selection approach. Hence, the algorithm converges much faster as compared to other approaches. As a result of Proposition2, $\mathbf{C}$ can be updated over partitions of $\mathbf{X}$, since for non-empty discrete sets $\mathcal{X}1$ and $\mathcal{X}2$ (Mair et al., 2017)

$$\operatorname{conv}(\mathcal{X}1 \cup \mathcal{X}2) = \operatorname{conv}(\operatorname{conv}(\mathcal{X}1) \cup \operatorname{conv}(\mathcal{X}2)). \quad (10)$$

Nevertheless, an alternative way is to adopt a sequential approach where instead of processing the whole collection of signals or $\mathbf{X}$, we process each signal (or a batch) sequentially one at a time, thereby jointly optimizing $\mathbf{C}$, $\mathbf{B}$ and $\mathbf{A}$ iteratively. In particular, we employed a sampling criterion based on which a new signal $\mathbf{x}_i$ is either included or not to a base dictionary $\mathbf{P}$ such that $\mathbf{D} = \mathbf{PB}$. Note that we have $\mathbf{P} = \mathbf{X}$ in standard AA model, and thus $\mathbf{P}$ should span the same space as $\mathbf{X}$, with $\mathbf{B}$ emphasizing the most important directions in that space. Due to inherent sparsity of $\mathbf{B}$, one can choose only the most important signals in $\mathbf{P}$ instead of the whole $\mathbf{X}$, thereby reducing the complexity of updating $\mathbf{C}$, and $\mathbf{B}$ significantly. Thus, we propose a sampling criterion which searches for the signals that does not lie in the span of the already selected signals. Specifically, any column $\mathbf{x}_i$ from $\mathbf{X}$ is added in $\mathbf{P}$ if the following condition holds

$$\|\mathbf{x}_i - \mathbf{\Pi}\mathbf{x}_i\|_2^2 = \|\mathbf{x}_i - \mathbf{P}\mathbf{P}^{\dagger}\mathbf{x}_i\|_2^2 > \tau \quad (11)$$

It computes the distance of vector $\mathbf{x}_i$ to the space spanned by the set $\mathbf{P}$. Here, $\mathbf{\Pi}$ is the projection matrix, $\mathbf{P}\mathbf{P}^{\dagger}\mathbf{x}_i$ is the projection of $\mathbf{x}_i$ on to $\mathbf{P}$, † denotes the pseudo-inverse and τ denotes a threshold value. Our implementation is inspired by the fast exemplar selection (FES) algorithm (see (Abrol et al., 2017) for more details) such as block matrix updates and incremental Cholesky factor updates.

8. Experimental Results and Comparison with Other Algorithms

This section studies the efficiency of the proposed GAA algorithm along with existing algorithms i.e., AA using active-set (AAAS) algorithm (Chen et al., 2014), AA using projected gradient (AAPG) algorithm (Mørup & Hansen, 2012), AA with Kullback-Leibler divergence (AAKL) algorithm (Diment & Virtanen, 2015), in various signal processing/machine learning tasks. For a fair comparison we have excluded the approaches such as (Mair & Brefeld, 2019) which apply AA on precomputed subset of data, as for very large number of examples any of the existing AA algorithms can be complemented with such approaches.

8.1. Synthetic Dataset

In this experiment, we considered the synthetic 2D-data of 1000 observations with the dimensionality of simplex set to 3 and 4, respectively. The various archetypes found by different algorithms are shown in Figure 2. It can be observed that for model order-3 GAA, AAAS and AAPG algorithms found archetypes which are lying in close vicinity (see Figure 2(a)). The AAKL algorithm performs poorly among all. For model order-4, we observed a 4% relative lower reconstruction error for all the observations than the AAPG algorithm (see Figure 2(b)). Here, GAA seems to be favouring the points near the top-right edge. Since, robust analysis is not considered here, its difficult to argue either in favor of the solution obtained via GAA or AAPG algorithm. Hence, both the results can be considered for good approximation of data.

We deliberately considered the model order-3 dataset, as there are no true archetypes available (Mørup & Hansen, 2012). However, as discussed earlier one can use the relaxed

AA model to find them. As an illustration, we have shown the archetypes found by the proposed GAA algorithm for various values of δ in Figure 2(c). Finally, Figure 2(d) shows the results for the robust AA model, and it can be observed that the proposed GAA algorithm performs well in finding archetypes robust to outliers.

8.2. Digit Classification

The state-of-the-art results for this task using existing kernel-DL methods are reported in (Abrol et al., 2016; Van Nguyen et al., 2013). This experiment uses the kernel AA model for digit classification task on USPS dataset (Bache & Lichman, 2013), to see if similar performance could be achieved by using AA instead of DL. Following the strategy in (Van Nguyen et al., 2013), we used the generative classification approach, where a test example is classified to the class that give the smallest reconstruction error. We first concatenate dictionaries (consisting of dictionary atoms or archetypes) from all (say Q) classes as:

$$\ddot{\mathbf{D}}_f = [\ddot{\mathbf{D}}^1 \dots \ddot{\mathbf{D}}^Q] = [\phi(\mathbf{X}^1)\mathbf{B}^1 \dots \phi(\mathbf{X}^Q)\mathbf{B}^Q] \quad (12)$$

The final dictionary $\ddot{\mathbf{D}}_f$ is used to solve for the sparse/convex decomposition $\mathbf{a}_t = [\mathbf{a}_t^1, \dots, \mathbf{a}_t^Q]$ for a given test example $\mathbf{x}_t$. Finally, the error with respect to q^{th} class is computed as:

$$\begin{aligned} r_t^q &= \|\phi(\mathbf{x}_t) - \phi(\mathbf{X}^q)\mathbf{B}^q\mathbf{a}_t^q\|_2^2 \quad \forall_q \quad q = 1, \dots, Q \\ &= \mathcal{K}(\mathbf{x}_t, \mathbf{x}_t) - 2\mathcal{K}(\mathbf{x}_t, \mathbf{X}^q)\mathbf{B}^q\mathbf{a}_t^q \\ &+ \mathbf{a}_t^{qT}\mathbf{B}^{qT}\mathcal{K}(\mathbf{X}^q, \mathbf{X}^q)\mathbf{B}^j\mathbf{a}_t^q \end{aligned} \quad (13)$$

In order to have a fair comparison, all experiments are performed under similar conditions. Specifically, dictionary for each class using kernel-GAA (KGAA), kernel-AAAS (KAAAS), kernel-KSVD (KKSVD) (Van Nguyen et al., 2013) and kernel sparse greedy dictionary (KSGD) (Abrol et al., 2016) algorithm is learned with the following parameters: 300 atoms/archetypes, cardinality in SC step 5, polynomial kernel of degree 4, error tolerance $\epsilon = 10^{-4}$ and maximum iterations 200. With additional complexity of implementing a Kernel version of AAODL algorithm, a comparison is excluded for this experiment, although we expect similar performance. It can be observed from results reported in Table 2 that AA achieves comparative performance at par with DL based methods even in presence of additive noise, and advocate towards its uses as an alternative for such tasks. Further, AA has the advantage of reduced computational cost in testing, since computing a convex representation is much more faster than computing a sparse representation, and more research is required to exploit these gains in many other tasks.

Table 2. Comparison of classification accuracies on USPS dataset for different methods.

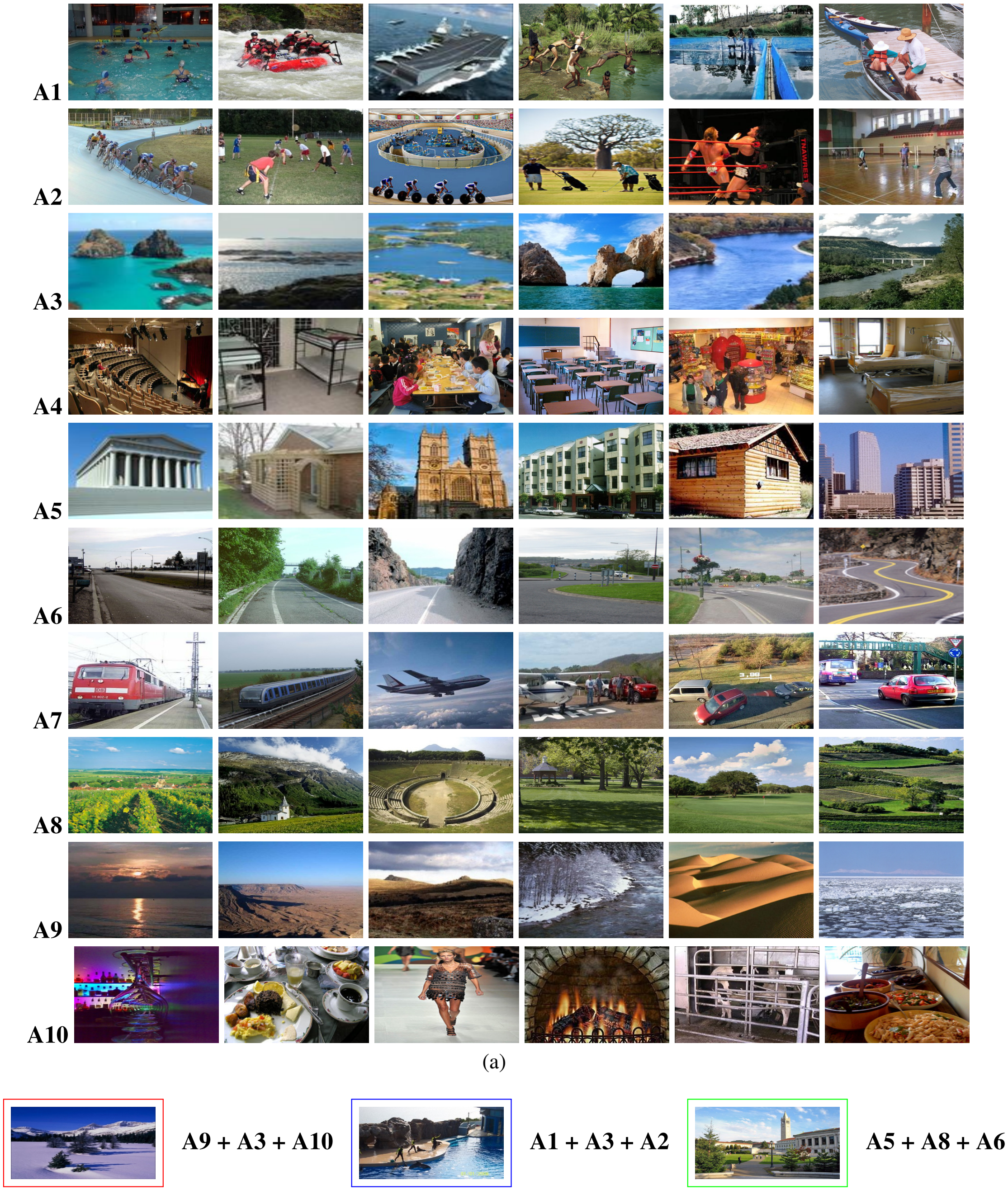
Method	Noise Standard Deviation σ				
	0	0.3	0.9	1.2	1.5
KKSVD	98.42	97.6	94.5	87.6	83.6
KSGD	98.40	97.5	94.4	87.6	83.6
KAAAS	98.36	97.3	94.2	87.3	83.8
KGAA	98.48	97.5	94.6	87.8	83.8

8.3. Archetypal Analysis In Large Image Collection

In this experiment, we analyze the SUN attribute image dataset: a subset of SUN image database (Xiao et al., 2010). The dataset consist of $14,340$ images and 102 attributes, and each image has been manually labelled with an attribute. For our analysis, we represented each image by a concatenation of GIST, HOG, and geometric context colour histogram features (see (Xiao et al., 2010) for more details), since they individually describe distinct visual phenomena. In the experiments reported here, we computed $1,000$ archetypes, which were further grouped into 10 categories (using spectral clustering) resembling to images of (i) water related activities such as sailing and swimming; (ii) physical activity such as sports, competition, and exercise; (iii) ocean, river and lake; (iv) enclosed area; (v) buildings and houses; (vi) highways and roads; (vii) transport (viii) open area with greenery; (ix) open area without greenery; and (x) abstract images. Figure 3(a) (columns $1-4$) shows few of the common generating images for each of the archetype, found by all AA algorithms. Figure 3(a), columns 5 and 6 shows example generating images found only by AAAS and GAA algorithms, respectively. It can be observed that the generating images are intuitively as expected. In addition, Figure 3(b) also shows few images, where each image is composed by three archetypes found by GAA algorithm. It can be observed that certain archetypes contribute structures while others are responsible for colour and intensity content. For instance, Green border image is composed of A5 (building), A8 (open area with greenery) and A6 (road).

9. Summary

In this paper, we presented a greedy algorithm for archetypal analysis. The proposed approach exploits the underlying sparsity pattern of the convex representations to identify archetypes or points on the convex hull of the data. This is done efficiently in the coefficient domain involving sparse matrices rather than the signal domain. The proposed method employs an iterative fast subset selection approach to find the archetypes. We have shown that the proposed algorithm has promising applications in computer



(a)

(b)

Figure 3. Visualization of the archetypes found for the SUN attribute dataset. (a) The top four generating images (columns 1-4) for each one of the ten archetypes (A1-A10), along with example images (column 5-6) found only by AAAS and GAA algorithm, respectively. (b) Example images with top three generating archetypes found by GAA algorithm.

vision/machine learning such as prediction tasks, and visualization for large databases of natural images. Further, we have shown that the proposed algorithm is also suitable for obtaining solutions to extended AA models.

References

Abrol, V., Sharma, P., and Sao, A. Greedy dictionary learning for kernel sparse representation based classifier. *Pattern Recognition Letters*, 78:64 – 69, 2016. ISSN 0167-8655. doi: 10.1016/j.patrec.2016.04.014.

Abrol, V., Sharma, P., and Sao, A. K. Fast exemplar selection algorithm for matrix approximation and representation: A variant oasis algorithm. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4436–4440, March 2017. doi: 10.1109/ICASSP.2017.7952995.

Adele Cutler, L. B. Archetypal analysis. *Technometrics*, 36 (4):338–347, November 1994. ISSN 00401706.

Arora, S., Ge, R., Kannan, R., and Moitra, A. Computing a nonnegative matrix factorization – provably. In *Annual ACM Symposium on Theory of Computing (STOC)*, pp. 145–162, New York, NY, USA, May 2012. doi: 10.1145/2213977.2213994.

Bache, K. and Lichman, M. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

Bauckhage, C. and Manshaei, K. Kernel archetypal analysis for clustering web search frequency time series. In *International Conference on Pattern Recognition (ICPR)*, pp. 1544–1549, August 2014. doi: 10.1109/ICPR.2014.274.

Bauckhage, C., Kersting, K., Hoppe, F., and Thurau, C. Archetypal analysis as an autoencoder. In *Workshop on New Challenges in Neural Computation*, pp. 8–15, October 2015.

Bernstein, D. *Matrix Mathematics: Theory, Facts, and Formulas (Second Edition)*. Princeton reference. Princeton University Press, 2009. ISBN 9780691140391.

Cabero, I. and Epifanio, I. Archetypal analysis: An alternative to clustering for unsupervised texture segmentation. *Image Analysis and Stereology*, 38(2):151–160, 2019. ISSN 1854-5165. doi: 10.5566/ias.2052.

Chen, W., Rodrigues, M., and Wassell, I. Projection design for statistical compressive sensing: A tight frame based approach. *IEEE Transactions on Signal Processing*, 61 (8):2016–2029, April 2013. ISSN 1053-587X. doi: 10.1109/TSP.2013.2245661.

Chen, Y., Mairal, J., and Harchaoui, Z. Fast and robust archetypal analysis for representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1478–1485, June 2014. ISBN 978-1-4799-5118-5. doi: 10.1109/CVPR.2014.192.

Chu, M. T. and Lin, M. M. Low-dimensional polytope approximation and its applications to nonnegative matrix factorization. *SIAM Journal on Scientific Computing*, 30 (3):1131–1155, 2008. doi: 10.1137/070680436.

Damle, A. and Sun, Y. A geometric approach to archetypal analysis and nonnegative matrix factorization. *Technometrics*, 59(3):361–370, 2017. doi: 10.1080/00401706.2016.1247017.

Das, A. and Kempe, D. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *International Conference on Machine Learning (ICML)*, pp. 1057–1064, June 2011. ISBN 978-1-4503-0619-5.

Diment, A. and Virtanen, T. Archetypal analysis for audio dictionary learning. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 1–5, October 2015. doi: 10.1109/WASPAA.2015.7336903.

Ding, W., Ishwar, P., and Saligrama, V. A provably efficient algorithm for separable topic discovery. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):712–725, June 2016. ISSN 1941-0484. doi: 10.1109/JSTSP.2016.2555240.

Elhamifar, E., Sapiro, G., and Vidal, R. See all by looking at a few: Sparse modeling for finding representative objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1600–1607, June 2012. doi: 10.1109/CVPR.2012.6247852.

Epifanio, I., Ibáñez, M. V., and Simó, A. Archetypal analysis with missing data: See all samples by looking at a few based on extreme profiles. *The American Statistician*, 0 (0):1–28, 2019. doi: 10.1080/00031305.2018.1545700.

Eugster, M. J. and Leisch, F. Weighted and robust archetypal analysis. *Computational Statistics and Data Analysis*, 55(3):1215 – 1225, May 2011. ISSN 0167-9473. doi: 10.1016/j.csda.2010.10.017.

Fotiadou, E., Panagakis, Y., and Pantic, M. Temporal archetypal analysis for action segmentation. In *IEEE International Conference on Automatic Face Gesture Recognition*, pp. 490–496, May 2017. doi: 10.1109/FG.2017.66.

Hinrich, J. L., Bardenfleth, S. E., Røge, R. E., Churchill, N. W., Madsen, K. H., and Mørup, M. Archetypal analysis for modeling multisubject fMRI data. *IEEE Journal of Selected Topics in Signal Processing*, 10 (7):1160–1171, October 2016. ISSN 1932-4553. doi: 10.1109/JSTSP.2016.2595103.

Hurley, N. and Rickard, S. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741, October 2009. doi: 10.1109/TIT.2009.2027527.

Jafari, M. and Plumbley, M. Fast dictionary learning for sparse representations of speech signals. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):1025–1031, September 2011. ISSN 1932-4553. doi: 10.1109/JSTSP.2011.2157892.

Javadi, H. and Montanari, A. Nonnegative matrix factorization via archetypal analysis. *Journal of the American Statistical Association*, 0(0):1–22, 2019. doi: 10.1080/01621459.2019.1594832.

Krause, A., Singh, A., and Guestrin, C. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, June 2008. ISSN 1532-4435.

Mair, S. and Brefeld, U. Coresets for archetypal analysis. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 7245–7253, December 2019.

Mair, S., Boubekki, A., and Brefeld, U. Frame-based data factorizations. In *International Conference on Machine Learning (ICML)*, pp. 2305–2313, August 2017.

McCallum, D. and Avis, D. A linear algorithm for finding the convex hull of a simple polygon. *Information Processing Letters*, 9(5):201 – 206, December 1979. ISSN 0020-0190. doi: 10.1016/0020-0190(79)90069-3.

Mei, J., Wang, C., and Zeng, W. Online dictionary learning for approximate archetypal analysis. In *European Conference on Computer Vision (ECCV)*, pp. 501–516, September 2018.

Moliner, J. and Epifanio, I. Robust multivariate and functional archetypal analysis with application to financial time series analysis. *Physica A: Statistical Mechanics and its Applications*, 519:195 – 208, 2019. ISSN 0378-4371. doi: https://doi.org/10.1016/j.physa.2018.12.036.

Mørup, M. and Hansen, L. K. Archetypal analysis for machine learning and data mining. *Neurocomputing: Special Issue on Machine Learning for Signal Processing*, 80:54 – 63, March 2012. ISSN 0925-2312. doi: 10.1016/j.neucom.2011.06.033.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, December 1978. ISSN 1436-4646. doi: 10.1007/BF01588971.

Seth, S. and Eugster, M. Archetypal analysis for nominal observations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(5):849–861, May 2016a. ISSN 0162-8828. doi: 10.1109/TPAMI.2015.2470655.

Seth, S. and Eugster, M. J. A. Probabilistic archetypal analysis. *Machine Learning*, 102(1):85–113, January 2016b. ISSN 1573-0565. doi: 10.1007/s10994-015-5498-8.

Thakur, A., Abrol, V., Sharma, P., and Rajan, P. Local compressed convex spectral embedding for bird species identification. *The Journal of the Acoustical Society of America*, 143(6):3819–3828, 2018. doi: 10.1121/1.5042241.

Thurau, C. and Bauckhage, C. Archetypal images in large photo collections. In *IEEE International Conference on Semantic Computing (ICSC)*, pp. 129–136, September 2009. doi: 10.1109/ICSC.2009.34.

Thurau, C., Kersting, K., Wahabzada, M., and Bauckhage, C. Convex non-negative matrix factorization for massive datasets. *Knowledge and Information Systems*, 29(2):457–478, November 2011. doi: 10.1007/s10115-010-0352-6.

Tosic, I. and Frossard, P. Dictionary learning. *IEEE Signal Processing Magazine*, 28(2):27–38, March 2011. ISSN 1053-5888. doi: 10.1109/MSP.2010.939537.

Van Nguyen, H., Patel, V., Nasrabadi, N., and Chellappa, R. Design of non-linear kernel dictionaries for object recognition. *IEEE Transactions on Image Processing*, 22 (12):5123–5135, December 2013. doi: 10.1109/TIP.2013.2282078.

Wynen, D., Schmid, C., and Mairal, J. Unsupervised learning of artistic styles with archetypal style analysis. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6584–6593, 2018.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3485–3492, June 2010. doi: 10.1109/CVPR.2010.5539970.

Yale Song, Vallmitjana, J., Stent, A., and Jaimes, A. TVSum: summarizing web videos using titles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5179–5187, June 2015. doi: 10.1109/CVPR.2015.7299154.

Zhao, C., Zhao, G., and Jia, X. Hyperspectral image unmixing based on fast kernel archetypal analysis. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, PP(99):1–16, September 2016. ISSN 1939-1404. doi: 10.1109/JSTARS.2016.2606504.

Zhao, G., Zhao, C., and Jia, X. Multilayer unmixing for hyperspectral imagery with fast kernel archetypal analysis. *IEEE Geoscience and Remote Sensing Letters*, 13(10):1532–1536, October 2016. doi: 10.1109/LGRS.2016.2595102.

Zhou, T., Bilmes, J., and Guestrin, C. Divide-and-conquer learning by anchoring a conical hull. In *International Conference on Neural Information Processing Systems (NIPS)*, pp. 1242–1250, December 2014.