A. Proofs

Proposition 1: Note that minimizing the cost function in (2) is equivalent to maximizing its sub-modular surrogate function

$$\mathcal{F}(\mathbf{X}, \mathbf{D}) = \sum_i \max_{|\Lambda_i| \leq k} |\langle \mathbf{D}_{\Lambda_i}, \mathbf{x}_i \rangle| \tag{14}$$

where Λ denotes the support set and k denotes the cardinality (Krause et al., 2008). The motivation for the proposed approach comes from a key structure of $\mathcal{F}()$ i.e., approximate sub-modularity. It has been shown that, this structure also relate to incoherence μ, which is a geometric property of the candidate training set (Nemhauser et al., 1978). Let's denote the candidate training signal set by $\mathcal{T}$, the selected and optimal dictionary atom set by $\mathcal{A}$ and $\mathcal{A}^*$, respectively. It has been shown that for submodular functions, a simple greedy algorithm that starts with an empty set $\mathcal{A} = \emptyset$, and iteratively adds a new element as:

$$u = \underset{u \in \mathcal{T} \setminus \mathcal{A}}{\operatorname{argmax}} \mathcal{F}(\mathcal{A} \cup u), \tag{15}$$

obtains a near-optimal solution with guarantee:

$$\mathcal{F}(\mathcal{A}) \geq e(\max \mathcal{F}(\mathcal{A}^*) - e^{'} \mu), \tag{16}$$

where the constants e and $e^{'}$ depends on the sparsity of the representation, and the second term is zero for monotonic submodular functions (Krause et al., 2008). This completes the proof for our claim that estimation of $\mathcal{A}$ is at-least a constant fraction of the optimal value (Krause et al., 2008; Nemhauser et al., 1978). One can derive even stronger multiplicative bounds by using the concept of submodularity ratio, recently developed in (Das & Kempe, 2011).

Proposition 2: Since archetypes lie on the boundary of the convex hull, it is possible to restrict the search of the archetypes to the points around it. Let us denote the points on boundary indexed by set $\mathcal{B}$, i.e., points that cannot be represented as convex combinations of other points except themselves. Note that no more than $n + 1$ extremal points are needed to represent a point (Mørup & Hansen, 2012). Hence, each $\mathbf{c}_i$ is computed (via non-negative least squares) iteratively by choosing one point after another and terminating with at most $n + 1$ positive values giving a sparse convex combination. In fact one can show that the positive values of the solution refer to points around the convex hull. It follows, since solving (4) for each point requires maximizing the negative gradient with respect to its representation vector $\mathbf{c}$,

$$-\nabla = \mathbf{X}^T(\mathbf{x} - \mathbf{X}\mathbf{c}) = \langle \mathbf{x}, \mathbf{x}_j \rangle - \sum_l c_l . \langle \mathbf{x}_l, \mathbf{x}_j \rangle, \tag{17}$$

and by property of linearity and convexity, inner product is maximized when one of the points is an extremal point (Mair et al., 2017). As a result solution to (4) ensures that union of the non-negative values of each $\mathbf{c}_i$ refers to points around the boundary. In fact with this reduced index set ($\mathcal{B}$) comprising of points only around the convex hull, we get the lossless non-negative factorization $\mathbf{X} = \mathbf{X}\mathbf{C} = \mathbf{X}[:, \mathcal{B}]\mathbf{C}[\mathcal{B}, :]$. This is the motivation behind our idea which presents a way to update archetypes in the coefficient domain using simple subset selection algorithm.

B. An alternative solution via ADMM algorithm

Another line of investigation is to solve (2) by ADMM algorithm by introducing an augmented variable similar to (4). However, this will make the problem more complex/time consuming. Our idea to introduce (4) is to exploit the underlying geometric property by segregating (2) into two simpler problems which can be solved separately. Note that there are multiple solutions for $\mathbf{C}$ which accounts for how signals from a subspace utilize one another in their convex representations. And the extremal points of interest are the ones having the sparsest representation. Also, (4) is a QP problem, solved using active set algorithm which has been shown to converge.

C. Algorithms for AA vs NMF

Non-negative matrix factorization (NMF) a well developed field, closely resembles AA, as in both cases the resulting representations are non-negative, and often one might think of solving AA via NMF style algorithms. However, note that while NMF only deals with non-negative data, AA can be applied to any matrix. Also, the dictionary in case of AA has a convex decomposition in data itself. Convexity in addition to just non-negativity results in more compact and meaning full decompositions. Recently, there has been attempts to adapt existing NMF based algorithms to solve AA like factorization problems. For instance, work in (Thurau et al., 2011) proposed an approach for convex-NMF where the representation are restricted to be convex combinations of data points.

While solving for a NMF decomposition exactly itself is NP-hard, (Arora et al., 2012). showed that exact solution can be obtained for separable matrix. Based upon this work, (Damle & Sun, 2017) extended NMF algorithm to AA for separable matrices. Geometrically this means that the dictionary in case of NMF (or archetypes in case of AA) are extremal data points of a polytope containing all data. If we force the archetypes to be data points then matrix $\mathbf{B}$ consist of a subset of rows of the identity matrix, and the problem becomes even combinatorially harder than the conventional AA. The solution presented by (Damle & Sun, 2017) revolves around to an extreme point finding problem which

Table 3. Comparison of mean (std) residual sum of squares error on Wine dataset for different methods over 100 trials.

Method	MCAR (10%)	MCAR (30%)	MAR	MNAR
IMP	6.583 (0.111)	8.172 (0.260)	7.315 (0.137)	8.223 (0.066)
MAAPG	6.601 (0.109)	8.130 (0.524)	6.800 (0.259)	8.178 (0.060)
MGAA	6.561 (0.121)	8.101 (0.345)	6.791 (0.162)	8.159 (0.060)
ORG	5.824 (0)	5.824 (0)	5.824 (0)	5.824 (0)
IMP1	5.985 (0.016)	6.011 (0.032)	6.894 (0.020)	6.150 (0.032)
MAAPG1	6.006 (0.015)	6.015 (0.028)	6.028 (0.021)	6.178 (0.027)
MGAA1	5.912 (0.017)	5.961 (0.027)	5.990 (0.014)	6.101 (0.029)

is an entire field of research in itself e.g, see (Zhou et al., 2014; Ding et al., 2016) and references within. Again, there is no free lunch i.e., even if we cast the problem as selecting suitable extremal points, we are still dealing with a subset selection problem of the order of $\binom{l}{d}$. Further, in practice we do not know whether a data matrix is separable and the archetypes might not necessarily be non-negative.

In a reverse direction, recent work in (Javadi & Montanari, 2019) used ideas from AA to come up with an algorithm for NMF based on Proximal Alternating minimization, when data matrix is non separable. The problem formulation is very similar to the relaxed-AA model discussed in our work where true archetypes doesn't exist.

D. Additional Experiments with Missing Data

Recently, there has been attempts to extend AA for missing data e.g., (Epifanio et al., 2019) proposed procedures to adapt existing algorithms by using missing value imputation or by projecting dissimilarities between samples. Similar to their experimental study we considered the Wine Dataset from UCI ML repository with three types of corruption: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). We compare the modified version of our GAA algorithm (MGAA) with the modified AAPG (MAAPG) and IMP (multiple imputation using additive regression, bootstrapping and predictive mean matching) as described in (Epifanio et al., 2019). For MGAA, we use imputation to modify the objective in (4) to $\|\mathbf{X} - \mathbf{W} \odot \mathbf{XC}\|_F^2$, where $\odot$ denotes element wise multiplication. Here, matrix $\mathbf{W}$ contains zeros in the positions where there are missing values in $\mathbf{X}$, ones in the column q (with n_q missing values) if there are not any missing values in $\mathbf{X}$ for that column, and $n/(n - n_q)$ otherwise. This ensure $\mathbf{C}$ doesn't have missing value, and similar procedure is used while updating $\mathbf{A}$ and $\mathbf{D}$.

A summary (mean and standard deviation) of the reconstruction error for each approach is reported in Table3. Since, the original datasets is available, we have also reported the reconstruction error on original data using the archetypes learned on original data denoted by ORG, and the archetypes learned from missing data via different approaches denoted by IMP1, MAAPG1 and MGAA1, respectively. It can be observed that MGAA performs better/comparable to existing approaches. Note that while here we have considered a simple imputation approach to handle missing entries, one can expect better gains using other ideas from say literature on matrix completion problem.