ISE

Industrial and
Systems Engineering

Self-Representation Based Unsupervised Exemplar Selection in a Union of Subspaces

Chong You[1], Chi Li[2], Daniel P. Robinson[3], and René Vidal[4]

[1]Department of Electrical Engineering and Computer Science, University of California, Berkeley, USA

[2]Apple Inc, Cupertino, USA

[3]Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

[4]Department of Biomedical Engineering, The Johns Hopkins University, USA

ISE Technical Report 22T-017

LEHIGH
UNIVERSITY

Self-Representation Based Unsupervised Exemplar Selection in a Union of Subspaces

Chong You, Chi Li, Daniel P. Robinson, and René Vidal, *Fellow, IEEE*

Abstract—Finding a small set of representatives from an unlabeled dataset is a core problem in a broad range of applications such as dataset summarization and information extraction. Classical exemplar selection methods such as k-medoids work under the assumption that the data points are close to a few cluster centroids, and cannot handle the case where data lie close to a union of subspaces. This paper proposes a new exemplar selection model that searches for a subset that best reconstructs all data points as measured by the ℓ_1 norm of the representation coefficients. Geometrically, this subset best covers all the data points as measured by the Minkowski functional of the subset. To solve our model efficiently, we introduce a farthest first search algorithm that iteratively selects the worst represented point as an exemplar. When the dataset is drawn from a union of independent subspaces, our method is able to select sufficiently many representatives from each subspace. We further develop an exemplar based subspace clustering method that is robust to imbalanced data and efficient for large scale data. Moreover, we show that a classifier trained on the selected exemplars (when they are labeled) can correctly classify the rest of the data points.

Index Terms—Unsupervised exemplar selection, imbalanced data, large-scale data, subspace clustering

1 Introduction

The availability of large annotated datasets in computer vision, such as ImageNet, has led to many recent breakthroughs in object detection and classification using supervised learning techniques such as deep learning. However, as data sizes continue to grow, it has become difficult to annotate the data for training fully supervised algorithms. As a consequence, the development of unsupervised learning techniques that can learn from *unlabeled* datasets has become extremely important. In addition to the challenge introduced by the sheer volume of data, the number of data samples in unlabeled datasets usually varies widely for different classes. For example, a street sign database collected from street view images may contain drastically different numbers of instances for different types of signs since not all of them are used on streets with the same frequency; a handwritten letter database may be highly imbalanced as the frequency of different letters in English text varies significantly (see Figure 1). An imbalanced data distribution is known to compromise performance of canonical supervised [1] and unsupervised [2] learning techniques.

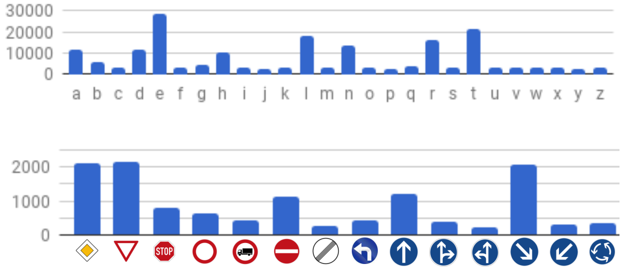


Fig. 1. Number of points in each class associated with the EMNIST handwritten letters (top) and the GTSRB (bottom) street sign databases.

We exploit the idea of exemplar selection to address the challenge of learning from an unlabeled dataset. Exemplar selection refers to the problem of selecting a set of data representatives or exemplars from the data. It has been a particularly useful approach for scaling up existing data clustering algorithms so that they can handle large datasets more efficiently [3]. Finding an exemplar set that is informative of the entire data is often the key challenge for the success of such approaches. Particularly, when the data is drawn from several different groups, it is crucial that an algorithm selects enough samples from each of the groups without prior knowledge of which points belong to which groups. This can be especially difficult when the data is imbalanced, as it is more likely to select data from over-represented groups than from under-represented groups.

Exemplar selection is also useful when one has limited resources so that only a small subset of data can be labeled. In such cases, exemplar selection can determine the subset to be manually labeled, and then used to train a model to infer labels for the remaining data [4]. The ability to correctly classify as many of the unlabeled data points as possible depends critically on the quality of the selected exemplars.

Some of the most popular methods for exemplar selection include k-centers and k-medoids, which search for the set of centers and medoids that best fit the data under the assumption that data points concentrate around a few discrete points. However, certain high-dimensional image and

- *C. You is with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, USA. Most of the work was done while C. You was at The Johns Hopkins University, USA.*
E-mail: cyou@berkeley.edu.
- *C. Li is a machine learning scientist in Apple Inc, Cupertino, USA.*
E-mail: chi_li@jhu.edu.
- *D. Robinson is with the Department of Industrial and Systems Engineering, Lehigh University, USA.*
E-mail: daniel.p.robinson@gmail.edu.
- *R. Vidal is with the Department of Biomedical Engineering, The Johns Hopkins University, USA.*
E-mail: rvidal@cis.jhu.edu.

video data is distributed among certain low-dimensional subspaces [5], [6], and the discrete center based methods become ineffective. *In this paper, we consider exemplar selection under a model where the data points lie close to a collection of unknown low-dimensional subspaces.* One line of work that can address such problem is based on the assumption that each data point can be expressed by a few data representatives with small reconstruction residual. This includes the simultaneous sparse representation [7] and dictionary selection [8], [9], which use greedy algorithms to solve their respective optimization problems, and group sparse representative selection [10], [11], [12], [13], [14], [15], which uses a convex optimization approach based on group sparsity. In particular, the analysis in [12] shows that when data come from a union of subspaces, their method is able to select a few representatives from each of the subspaces. However, methods in this category cannot effectively handle large-scale data as they have quadratic complexity in the number of points. Moreover, the convex optimization based methods such as that in [12] are not flexible in selecting a desired number of representatives since the size of the subset cannot be directly controlled by adjusting an algorithm parameter.

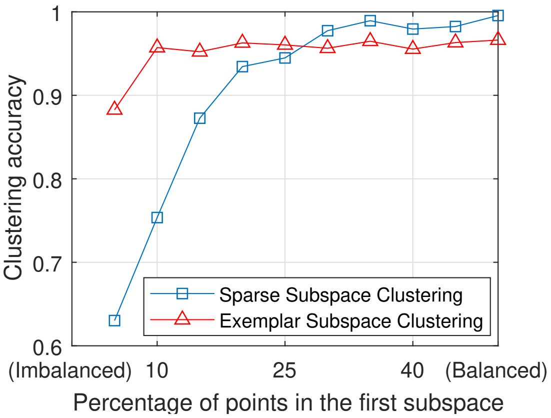


Fig. 2. Subspace clustering on imbalanced data. Two subspaces of dimension three are generated uniformly at random in ambient space of dimension five. Then, x and $100 - x$ points are sampled uniformly at random from the two subspaces, respectively, where x is varied in the x-axis. The clustering accuracy of SSC decreases dramatically as the dataset becomes imbalanced. The exemplar based subspace clustering (see Algorithm 3) is more robust to imbalanced data distribution.

1.1 Paper contributions

We present a data self-representation based exemplar selection algorithm for learning from large scale and imbalanced data in an unsupervised manner. Our method is based on the *self-expressiveness* property of data in a union of subspaces [16], which states that each data point in a union of subspaces can be written as a linear combination of other points from its own subspace. That is, given data $\mathcal{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\} \subseteq I\!R^D$, there exists $\{c_{ij}\}$ such that $\boldsymbol{x}_j = \sum_{i \neq j} c_{ij} \boldsymbol{x}_i$ and c_{ij} is nonzero only if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are from the same subspace. Such representations $\{c_{ij}\}$ are called *subspace-preserving*. In particular, if the subspace dimensions are small, then the representations can be taken to be sparse. Based on this observation, [16] proposes the Sparse Subspace Clustering (SSC) method, which computes for each $\boldsymbol{x}_j \in \mathcal{X}$ the vector $\boldsymbol{c}_j = [c_{1j}, \cdots, c_{Nj}]^\top$ as a solution to the sparse optimization problem

$$\min_{\boldsymbol{c} \in \mathbb{R}^N} \|\boldsymbol{c}\|_1 + \tfrac{\lambda}{2} \|\boldsymbol{x}_j - \sum_{i \neq j} c_i \boldsymbol{x}_i\|_2^2, \tag{1}$$

where $\lambda > 0$. In [16], the solution to (1) is used to define an affinity between any pair of points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as $|c_{ij}| + |c_{ji}|$, and then spectral clustering is applied to generate a segmentation of the data points into their respective subspaces. Existing theoretical results show that, under certain assumptions on the data, the solution to (1) is subspace-preserving [17], [18], [19], [20], [21], [22], [23], [24], [25], thus justifying the correctness of the affinity produced by SSC.

While the nonzero entries for each $\boldsymbol{c}_j$ determine a subset of $\mathcal{X}$ that can represent $\boldsymbol{x}_j$ with the minimum ℓ_1-norm on the coefficients, the union of the representations over all $\{\boldsymbol{c}_j\}$ often uses the entire dataset $\mathcal{X}$. In this paper, we propose to find a small subset $\mathcal{X}_0 \subseteq \mathcal{X}$, which we call *exemplars*, such that solutions $\boldsymbol{c}_j$ to the problem

$$\min_{\boldsymbol{c} \in \mathbb{R}^N} \|\boldsymbol{c}\|_1 + \tfrac{\lambda}{2} \|\boldsymbol{x}_j - \sum_{i: \boldsymbol{x}_i \in \mathcal{X}_0} c_i \boldsymbol{x}_i\|_2^2 \tag{2}$$

are also subspace-preserving. Since $\mathcal{X}_0$ is a small subset of $\mathcal{X}$, solving the optimization problem (2) is much cheaper computationally compared to (1). Computing an appropriate $\mathcal{X}_0$ through an exhaustive search would be computationally impractical. To address this issue, we present an efficient algorithm (an exemplar selection algorithm) that iteratively selects the worst represented point from the data $\mathcal{X}$ to form $\mathcal{X}_0$. Our exemplar selection procedure is then used to design an exemplar-based subspace clustering approach (assuming that the exemplars are unlabeled) [26] and an exemplar-based classification approach (assuming that the exemplars are labeled) by using the representative power of the selected exemplars. In summary, our work makes the following contributions compared to the state of the art:

- We present a geometric interpretation of our exemplar selection algorithm as one of finding a subset of the data that *best covers* the entire dataset as measured by the Minkowski functional of the subset. When the data lies in a union of independent subspaces, we prove that our method selects sufficiently many representative data points (exemplars) from each subspace, even when the dataset is imbalanced. Unlike prior methods such as [12], our method has linear execution time and memory complexity in the number of data points for each iteration, and can be terminated when the desired number of exemplars have been selected.
- We show that the exemplars in $\mathcal{X}_0$ selected by our method can be used for subspace clustering by first computing the representations for each data point with respect to the exemplars as in (2), second constructing a k-nearest neighbor graph of the representation vectors, and third applying spectral clustering. Compared to SSC, the exemplar-based subspace clustering method is empirically less sensitive to imbalanced data and more efficient on large-scale datasets (see Figure 2). Experimental results on the large-scale and label-imbalanced handwritten letter dataset EMNIST and street sign dataset GTSRB show that our method outperforms state-of-the-art algorithms in terms of both

clustering performance and running time.

- We show that a classifier trained on the exemplars selected by our model (assuming that the labels of the exemplars are provided) is able to correctly classify the rest of the data points. We demonstrate through experiments on the Extended Yale B face database that exemplars selected by our method produce higher classification accuracy when compared to several popular exemplar selection methods.

We remark that a conference version of the paper appeared in the proceedings of European Conference on Computer Vision (ECCV) in 2018 [26]. In comparison to the conference version, which focuses on the problem of subspace clustering on imbalanced data, the current paper addresses the problem of exemplar selection, which has a broader range of applications that include data summarization, clustering and classification tasks. With additional technical results and experimental evaluation, the current paper provides a more comprehensive study of the subject.

1.2 Related work

Exemplar selection. Two of the most popular methods for exemplar selection are k-centers and k-medoids. The k-centers problem is a data clustering problem studied in theoretical computer science and operations research. Given a set $\mathcal{X}$ and an integer k, the goal is to find a set of centers $\mathcal{X}_0 \subseteq \mathcal{X}$ with $|\mathcal{X}_0| \leq k$ that minimizes the quantity $\max_{\boldsymbol{x}\in\mathcal{X}} d^2(\boldsymbol{x}, \mathcal{X}_0)$, where $d^2(\boldsymbol{x}, \mathcal{X}_0) := \min_{\boldsymbol{v}\in\mathcal{X}_0} \|\boldsymbol{x} - \boldsymbol{v}\|_2^2$ is the squared distance of $\boldsymbol{x}$ to the closest point in $\mathcal{X}_0$. A partition of $\mathcal{X}$ is given by the closest center to which each point $\boldsymbol{x} \in \mathcal{X}$ belongs. The k-medoids is a variant of k-centers that minimizes the sum of the squared distances, i.e., minimizes $\sum_{\boldsymbol{x}\in\mathcal{X}} d^2(\boldsymbol{x}, \mathcal{X}_0)$ instead of the maximum distance. However, both k-centers and k-medoids model data as concentrating around several cluster centers, and do not generally apply to data lying in a union of subspaces.

In general, selecting a representative subset of the entire data has been studied in a wide range of contexts such as Determinantal Point Processes [27], [28], [29], Prototype Selection [30], [31], Rank Revealing QR [32], Column Subset Selection (CSS) [33], [34], [35], [36], separable Nonnegative Matrix Factorization (NMF) [37], [38], [39], and so on [40]. In particular, both CSS and separable NMF can be interpreted as finding exemplars such that each data point can be expressed as a linear combination of such exemplars. However, these methods do not impose sparsity on the representation coefficients, and therefore cannot be used to select good representatives from data that is drawn from a union of low-dimensional subspaces.

Subspace clustering on imbalanced and large scale data. Subspace clustering aims to cluster data points drawn from a union of subspaces into their respective subspaces. Recently, self-expressiveness based subspace clustering methods such as SSC and its variances [41], [42], [43], [44], [45], [46], [47] have achieved great success for many computer vision tasks such as face clustering, handwritten digit clustering, and so on. Nonetheless, previous experimental evaluations focused primarily on balanced datasets, i.e. datasets with approximately the same number of samples from each cluster. In practice, datasets are often imbalanced and such skewed data distributions can significantly compromise the clustering performance of SSC. There has been no study of this issue in the literature to the best of our knowledge.

Another issue with many self-expressive based subspace clustering methods is that they are limited to small or medium scale datasets [48]. Several works addressed the scalability issue by computing a dictionary with number of atoms much smaller than the total number of data points in $\mathcal{X}$, and expressing each data point in $\mathcal{X}$ as a linear combination of the atoms in the dictionary (the dictionary is usually not a subset of $\mathcal{X}$). In particular, [49] shows that if the atoms in the dictionary happen to lie in the same union of subspaces as the input data $\mathcal{X}$, then this approach is guaranteed to be correct. However, there is little evidence that such a condition is satisfied for real data as the atoms of the dictionary are not constrained to be a subset of $\mathcal{X}$. Another recent work [50], which uses data-independent random matrices as dictionaries, also suffers from this issue and lacks correctness guarantees. More recently, several works [51], [52], [53] use exemplar selection to form the dictionary for subspace clustering, but they lack theoretical justification that their selected exemplars represent the subspaces.

2 Self-Representation based Unsupervised Exemplar Selection

In this section, we present our self-representation based method for exemplar selection from an unlabeled dataset $\mathcal{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$, which are assumed to have unit ℓ_2 norm.[1] We first formulate the model for selecting a subset $\mathcal{X}_0$ of exemplars from $\mathcal{X}$ in Section 2.1 as minimizing a self-representation cost. Since the model is a combinatorial optimization problem, we present an efficient algorithm for solving it approximately in Section 2.2.

2.1 A self-representation cost for exemplar selection

In our exemplar selection model, the goal is to find a small subset $\mathcal{X}_0 \subseteq \mathcal{X}$ that can linearly represent all data points in $\mathcal{X}$. In particular, the set $\mathcal{X}_0$ should contain exemplars from each subspace such that the solution to (2) for each data point $\boldsymbol{x}_j \in \mathcal{X}$ is subspace-preserving. Next, we define a cost function based on the optimization problem in (2) and then present our exemplar selection model.

Definition 1 (Self-representation cost function)**.** Given $\mathcal{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\} \subseteq \mathbb{R}^D$, we define the self-representation cost function $F_\lambda : 2^{\mathcal{X}} \to \mathbb{R}$ as

$$F_\lambda(\mathcal{X}_0) := \sup_{\boldsymbol{x}_j\in\mathcal{X}} f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0) \tag{3}$$

where

$$f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0) := \min_{\boldsymbol{c}\in\mathbb{R}^N} \|\boldsymbol{c}\|_1 + \frac{\lambda}{2}\|\boldsymbol{x}_j - \sum_{i:\boldsymbol{x}_i\in\mathcal{X}_0} c_i\boldsymbol{x}_i\|_2^2 \tag{4}$$

and $\lambda \in (1, \infty)$ is a parameter. By convention, we define $f_\lambda(\boldsymbol{x}_j, \emptyset) = \frac{\lambda}{2}$ for all $\boldsymbol{x}_j \in \mathcal{X}$, where $\emptyset$ is the empty set.

The quantity $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0)$ is a measure of how well the data point $\boldsymbol{x}_j \in \mathcal{X}$ is represented by the subset $\mathcal{X}_0$. The function $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0)$ has the following properties.

[1] This is not a strong assumption as one can always normalize the data points as a preprocessing step for any given dataset.

Lemma 1. *For each $j \in \{1, \dots, N\}$, the function $f_\lambda(\boldsymbol{x}_j, \cdot)$ is monotone with respect to the partial order defined by set inclusion, i.e., $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0') \geq f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0'')$ for any $\emptyset \subseteq \mathcal{X}_0' \subseteq \mathcal{X}_0'' \subseteq \mathcal{X}$.*

Proof. Let $j \in \{1, \dots N\}$. Then, let us define $\boldsymbol{c}'$ as

$$\boldsymbol{c}' = [c_1', \cdots, c_N']^\top \in \arg\min_{\boldsymbol{c} \in \mathbb{R}^N} \|\boldsymbol{c}\|_1 + \tfrac{\lambda}{2}\|\boldsymbol{x}_j - \sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0'} c_i \boldsymbol{x}_i\|_2^2.$$

It follows from the optimality conditions that $c_i' = 0$ for all i such that $\boldsymbol{x}_i \notin \mathcal{X}_0'$. Combining this with $\mathcal{X}_0' \subseteq \mathcal{X}_0''$ yields

$$\begin{aligned}
f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0') &= \|\boldsymbol{c}'\|_1 + \tfrac{\lambda}{2}\|\boldsymbol{x}_j - \sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0'} c_i' \boldsymbol{x}_i\|_2^2 \\
&= \|\boldsymbol{c}'\|_1 + \tfrac{\lambda}{2}\|\boldsymbol{x}_j - \sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0''} c_i' \boldsymbol{x}_i\|_2^2 \\
&\geq \min_{\boldsymbol{c} \in \mathbb{R}^N} \|\boldsymbol{c}\|_1 + \tfrac{\lambda}{2}\|\boldsymbol{x}_j - \sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0''} c_i \boldsymbol{x}_i\|_2^2 \\
&= f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0^{''}),
\end{aligned}$$

which is the desired result. □

Lemma 2. *For each $j \in \{1, \dots, N\}$ the following hold: (i) for every $\mathcal{X}_0 \in 2^{\mathcal{X}}$ the inclusion $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0) \in [1 - \frac{1}{2\lambda}, \frac{\lambda}{2}]$ holds; (ii) $f_\lambda(\boldsymbol{x}_j, \emptyset) = \lambda/2$; and (iii) $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0) = 1 - \frac{1}{2\lambda}$ if and only if at least one of $\boldsymbol{x}_j$ or $-\boldsymbol{x}_j$ is in $\mathcal{X}_0$.*

Proof. First observe that if $\mathcal{X}_0 = \emptyset$, then it follows from Definition 1 that $f_\lambda(\boldsymbol{x}_j, \emptyset) = \lambda/2$. Second, consider the case $\mathcal{X}_0 = \mathcal{X}$. In this case, define $\bar{\boldsymbol{c}} = [\bar{c}_1, \cdots, \bar{c}_N]$ to be the one-hot vector with j-th entry $\bar{c}_j = 1 - \frac{1}{\lambda}$ and all other entries zero. One can then verify that $\|\bar{\boldsymbol{c}}\|_1 + \frac{\lambda}{2}\|\boldsymbol{x}_j - \sum_{i=1}^N \bar{c}_i \boldsymbol{x}_i\|_2^2 = 1 - \frac{1}{2\lambda}$ (by recalling the assumption that $\|\boldsymbol{x}_j\|_2 = 1$). Combining these two cases with Lemma 1 establishes that parts (i) and (ii) hold.

For the "if" direction of part (iii), let either $\boldsymbol{x}_j \in \mathcal{X}_0$ or $-\boldsymbol{x}_j \in \mathcal{X}_0$. Define $\bar{\boldsymbol{c}} = [\bar{c}_1, \cdots, \bar{c}_N]$ as a one-hot vector with j-th entry $\bar{c}_j = 1 - \frac{1}{\lambda}$ if $\boldsymbol{x}_j \in \mathcal{X}_0$, and $\bar{c}_j = -(1 - \frac{1}{\lambda})$ if $-\boldsymbol{x}_j \in \mathcal{X}_0$; in either case all other entries are set to zero. One can then verify that $\|\bar{\boldsymbol{c}}\|_1 + \frac{\lambda}{2}\|\boldsymbol{x}_j - \sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0} \bar{c}_i \boldsymbol{x}_i\|_2^2 = 1 - \frac{1}{2\lambda}$, which completes the proof for this direction.

To prove the "only if" direction, suppose that $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0) = 1 - \frac{1}{2\lambda}$. Let us define

$$\boldsymbol{c}^* \in \arg\min_{\boldsymbol{c}} \|\boldsymbol{c}\|_1 + \tfrac{\lambda}{2}\|\boldsymbol{x}_j - \sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0} c_i \boldsymbol{x}_i\|_2^2$$

and $\boldsymbol{e}^* = \boldsymbol{x}_j - \sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0} c_i^* \boldsymbol{x}_i$. From the optimality conditions, it follows that $c_i^* = 0$ for all $i \in \{1, \cdots, N\}$ such that $\boldsymbol{x}_i \notin \mathcal{X}_0$. Using this fact, the assumption that the data is normalized, and basic properties of norms, we have

$$\begin{aligned}
1 = \|\boldsymbol{x}_j\|_2 &= \|\boldsymbol{e}^* + \sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0} c_i^* \boldsymbol{x}_i\|_2 \\
&\leq \|\boldsymbol{e}^*\|_2 + \|\sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0} c_i^* \boldsymbol{x}_i\|_2 \\
&\leq \|\boldsymbol{e}^*\|_2 + \sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0} (|c_i^*| \|\boldsymbol{x}_i\|_2) = \|\boldsymbol{e}^*\|_2 + \|\boldsymbol{c}^*\|_1.
\end{aligned} \tag{5}$$

From (5), $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0) = 1 - \frac{1}{2\lambda}$ and definition of $\boldsymbol{c}^*$, we have

$$\begin{aligned}
1 - \tfrac{1}{2\lambda} &= f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0) \qquad (6) \\
&= \|\boldsymbol{c}^*\|_1 + \tfrac{\lambda}{2}\|\boldsymbol{e}^*\|_2^2 \geq 1 - \|\boldsymbol{e}^*\|_2 + \tfrac{\lambda}{2}\|\boldsymbol{e}^*\|_2^2 \geq 1 - \tfrac{1}{2\lambda},
\end{aligned}$$

where the last inequality follows by computing the minimum value of $1 - \|\boldsymbol{e}^*\|_2 + \frac{\lambda}{2}\|\boldsymbol{e}^*\|_2^2$. It follows that equality is achieved for all inequalities in (6). By requiring equality for the second and first inequalities in (6), we get respectively,

$$\|\boldsymbol{e}^*\|_2 = \tfrac{1}{\lambda} \text{ and } \|\boldsymbol{c}^*\|_1 = 1 - \tfrac{1}{\lambda}. \tag{7}$$

Since (7) implies $\|\boldsymbol{e}^*\|_2 + \|\boldsymbol{c}^*\|_1 = 1$, we can conclude that all of the inequalities in (5) must actually be equalities. Using this fact and (5) we have that

$$\|\sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0} c_i^* \boldsymbol{x}_i\|_2 = 1 - \|\boldsymbol{e}^*\|_2 = 1 - \tfrac{1}{\lambda}. \tag{8}$$

Define $\mu_0 := \max_{i:\boldsymbol{x}_i \in \mathcal{X}_0} |\langle \boldsymbol{x}_j, \boldsymbol{x}_i \rangle|$. From definition of $\boldsymbol{e}^*$, (7), the fact that the data is normalized, and (8), we have

$$\begin{aligned}
\tfrac{1}{\lambda^2} = \|\boldsymbol{e}^*\|_2^2 &= \|\boldsymbol{x}_j - \sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0} c_i^* \boldsymbol{x}_i\|_2^2 \\
&= 1 - 2\langle \boldsymbol{x}_j, \sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0} c_i^* \boldsymbol{x}_i \rangle + (1 - \tfrac{1}{\lambda})^2. \qquad (9)
\end{aligned}$$

For the second term on the right hand side of (9), we may use the fact that the data is normalized, definition of μ_0, and (7) to conclude that

$$\langle \boldsymbol{x}_j, \sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0} c_i^* \boldsymbol{x}_i \rangle = \sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0} c_i^* \langle \boldsymbol{x}_j, \boldsymbol{x}_i \rangle \leq \mu_0 \|\boldsymbol{c}^*\|_1 = \mu_0(1 - \tfrac{1}{\lambda}).$$

Plugging this into (9) yields

$$\tfrac{1}{\lambda^2} \geq 1 - 2\mu_0(1 - \tfrac{1}{\lambda}) + (1 - \tfrac{1}{\lambda})^2, \tag{10}$$

which after simplification shows that

$$0 \geq -2\mu_0(1 - \tfrac{1}{\lambda}) + 2(1 - \tfrac{1}{\lambda}) = 2(1 - \tfrac{1}{\lambda})(1 - \mu_0). \tag{11}$$

Recall that $\lambda \in (1, \infty)$ (see Definition 1). Therefore, from (11) we see that $\mu_0 = \max_{i:\boldsymbol{x}_i \in \mathcal{X}_0} |\langle \boldsymbol{x}_j, \boldsymbol{x}_i \rangle| \geq 1$. Since both $\boldsymbol{x}_j$ and $\boldsymbol{x}_i$ have unit ℓ_2 norm, we conclude that $\mu_0 = 1$, i.e., that either $\boldsymbol{x}_j$ or $-\boldsymbol{x}_j$ must be in $\mathcal{X}_0$, as desired. □

Observe that if $\mathcal{X}_0$ contains enough exemplars from the subspace containing $\boldsymbol{x}_j$ and a solution $\boldsymbol{c}^*$ to the optimization problem in (4) is subspace-preserving, then it is expected that $\boldsymbol{c}^*$ will be sparse and that the residual $\boldsymbol{x}_j - \sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0} \boldsymbol{x}_i c_i^*$ will be close to zero. This suggests that we should select the subset $\mathcal{X}_0$ such that the value $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0)$ is small for all j. As the value $F_\lambda(\mathcal{X}_0)$ is achieved by the data point $\boldsymbol{x}_j$ that has the largest value $f(\boldsymbol{x}_j, \mathcal{X}_0)$, we propose to perform exemplar selection by searching for a subset $\mathcal{X}_0^* \subseteq \mathcal{X}$ that minimizes the self-representation cost function, i.e.,

$$\mathcal{X}_0^* = \arg\min_{|\mathcal{X}_0| \leq k} F_\lambda(\mathcal{X}_0), \tag{12}$$

where $k \in \mathbb{Z}$ is the target number of exemplars. The objective function $F_\lambda(\cdot)$ in (12) is monotone as shown next.

Lemma 3. *If $\emptyset \subseteq \mathcal{X}_0' \subseteq \mathcal{X}_0'' \subseteq \mathcal{X}$, then $F_\lambda(\mathcal{X}_0') \geq F_\lambda(\mathcal{X}_0'')$.*

Proof. Let us define

$$\boldsymbol{x}' \in \arg\sup_{\boldsymbol{x}_j \in \mathcal{X}} f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0') \text{ and } \boldsymbol{x}'' \in \arg\sup_{\boldsymbol{x}_j \in \mathcal{X}} f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0'').$$

It follows from these definitions and Lemma 1 that

$$\begin{aligned}
F_\lambda(\mathcal{X}_0') &= f_\lambda(\boldsymbol{x}', \mathcal{X}_0') \\
&\geq f_\lambda(\boldsymbol{x}'', \mathcal{X}_0') \geq f_\lambda(\boldsymbol{x}'', \mathcal{X}_0'') = F_\lambda(\mathcal{X}_0''),
\end{aligned}$$

which completes the proof. □

2.2 A Farthest First Search (FFS) algorithm

Solving the optimization problem (12) is NP-hard in general as it requires evaluating $F_\lambda(\mathcal{X}_0)$ for each subset $\mathcal{X}_0$ of size at most k. In Algorithm 1 below, we present a greedy algorithm for efficiently computing an approximate solution to (12). The algorithm progressively grows a candidate subset $\mathcal{X}_0$ (initialized as the empty set) until it reaches the desired size k. During each iteration i, step 3 of the algorithm selects the point $\boldsymbol{x}_j \in \mathcal{X}$ that is worst represented by the current subset $\mathcal{X}_0^{(i)}$ as measured by $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0^{(i)})$. It was shown in Lemma 2 that $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0^{(i)}) = 1 - \frac{1}{2\lambda}$ if $\boldsymbol{x}_j \in \mathcal{X}_0^{(i)}$, and $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0^{(i)}) > 1 - \frac{1}{2\lambda}$ if $\boldsymbol{x}_j \notin \mathcal{X}_0^{(i)}$ and $-\boldsymbol{x}_j \notin \mathcal{X}_0^{(i)}$. Thus, during each iteration an element not from $\mathcal{X}_0^{(i)}$ is added to $\mathcal{X}_0^{(i)}$ to form $\mathcal{X}_0^{(i+1)}$ when N is sufficiently large. When the algorithm terminates, the output $\mathcal{X}_0^{(k)}$ contains exactly k distinct exemplars from $\mathcal{X}$.

We also note that the FFS algorithm can be viewed as an extension of the farthest first traversal algorithm (see, e.g. [54]), which is an approximation algorithm for the k-centers problem discussed in Section 1.2.

Algorithm 1 A farthest first search (FFS) algorithm for exemplar selection

```
Input: Data $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\} \subseteq \mathbb{R}^D$, parameter $\lambda > 1$
    and number of desired exemplars $k \ll N$.
 1: Select $j \in \{1, \dots, N\}$ randomly and set $\mathcal{X}_0^{(1)} \leftarrow \{\boldsymbol{x}_j\}$.
 2: for $i = 1, \cdots, k-1$ do
 3:     $\mathcal{X}_0^{(i+1)} = \mathcal{X}_0^{(i)} \cup \arg\max_{\boldsymbol{x}_j \in \mathcal{X}} f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0^{(i)})$
 4: end for
Output: $\mathcal{X}_0^{(k)}$
```

Algorithm 2 An efficient implementation of FFS

```
Input: Data $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\} \subseteq \mathbb{R}^D$, parameters $\lambda > 1$
    and number of desired exemplars $k \ll N$.
 1: Select $j \in \{1, \dots, N\}$ randomly and set $\mathcal{X}_0^{(1)} \leftarrow \{\boldsymbol{x}_j\}$.
 2: Compute $b_j = f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0^{(1)})$ for $j = 1, \cdots, N$.
 3: for $i = 1, \cdots, k-1$ do
 4:     Let $o_1, \cdots, o_N$ be a permutation of $1, \cdots, N$ such that
                $b_{o_p} \geq b_{o_q}$  when  $p < q$.
 5:     Initialize $max\_cost = 0$.
 6:     for $j = 1, \cdots, N$ do
 7:         Set $b_{o_j} = f_\lambda(\boldsymbol{x}_{o_j}, \mathcal{X}_0^{(i)})$.
 8:         if $b_{o_j} > max\_cost$ then
 9:             Set $max\_cost = b_{o_j}$, $new\_index = o_j$.
10:         end if
11:         if $j = N$ or $max\_cost \geq b_{o_{j+1}}$ then
12:             break
13:         end if
14:     end for
15:     $\mathcal{X}_0^{(i+1)} = \mathcal{X}_0^{(i)} \cup \{\boldsymbol{x}_{new\_index}\}$.
16: end for
Output: $\mathcal{X}_0^{(k)}$
```

Efficient implementation. Observe that each iteration of Algorithm 1 requires evaluating $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0^{(i)})$ for every $\boldsymbol{x}_j \in \mathcal{X}$. Therefore, the complexity of Algorithm 1 is linear in the number of data points N assuming k is fixed and small. However, computing $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0^{(i)})$ itself is not easy as it requires solving a sparse optimization problem. Next, we introduce an efficient implementation of Algorithm 1 that accelerates the procedure by eliminating the need to compute $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0^{(i)})$ for some $\boldsymbol{x}_j$ in each iteration.

The idea underpinning the computational savings in Algorithm 2 is the monotonicity of $f_\lambda(\boldsymbol{x}_j, \cdot)$ (see Lemma 1). That is, for any $\emptyset \subseteq \mathcal{X}_0' \subseteq \mathcal{X}_0'' \subseteq \mathcal{X}$ we have $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0') \geq f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0'')$. Since in the FFS algorithm the set $\mathcal{X}_0^{(i)}$ is progressively increased, this implies that $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0^{(i)})$ is non-increasing in i. In step 2 we initialize $b_j = f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0^{(1)})$ for each $j \in \{1, \cdots, N\}$, which is an upper bound for $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0^{(i)})$ for $i \geq 1$. In each iteration i, the goal is to find a data point that maximizes $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0^{(i)})$. To do this, we first find an ordering $o_1, \cdots, o_N$ of $1, \cdots, N$ such that $b_{o_1} \geq \cdots \geq b_{o_N}$ (step 4). We then compute $f_\lambda(\cdot, \mathcal{X}_0^{(i)})$ sequentially for points in $\boldsymbol{x}_{o_1}, \cdots, \boldsymbol{x}_{o_N}$ (step 7) while tracking the highest value of $f_\lambda(\cdot, \mathcal{X}_0^{(i)})$ by the variable *max_cost* (step 9). Once the condition that *max_cost* $\geq b_{o_{j+1}}$ is met (step 11), we can assert that for any $j' > j$ the point $\boldsymbol{x}_{o_{j'}}$ is not a maximizer. This can be seen from $f_\lambda(\boldsymbol{x}_{o_{j'}}, \mathcal{X}_0^{(i)}) \leq b_{o_{j'}} \leq b_{o_{j+1}} \leq$ *max_cost*, where the first inequality follows from the monotonicity of $f_\lambda(\boldsymbol{x}_{o_{j'}}, \mathcal{X}_0^{(i)})$ as a function of i. Thus, we can break the loop (step 12) and avoid computing $f_\lambda(\boldsymbol{x}_{o_j}, \mathcal{X}_0^{(i)})$ for the remaining values of j in this iteration. When Algorithm 2 terminates, it produces the same output as Algorithm 1 but with a reduced total number of evaluations for $f_\lambda(\cdot, \cdot)$.

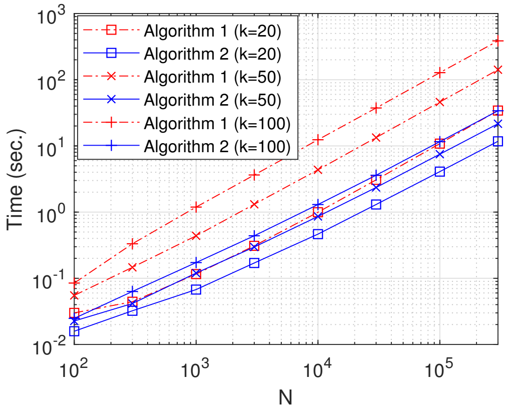


Fig. 3. Running time for Algorithm 1 and Algorithm 2 on a synthetically generated dataset where N data points are sampled uniformly at random from the unit sphere of $\mathbb{R}^{10}$ averaged over 10 trials. N is varied along the x-axis and takes values between 100 and $300{,}000$.

Figure 3 reports the computational time of Algorithm 1 and Algorithm 2 with synthetically generated data where N data points are sampled uniformly at random from the unit sphere of $\mathbb{R}^{10}$. It shows that the efficient implementation in Algorithm 2 is around 2 to 10 times faster than the naive implementation in Algorithm 1. Comparing the results across different values of k, we find that the benefit of Algorithm 2 is more prominent for larger values of k.

3 Theoretical Analysis

In this section, we study the theoretical properties of the self-representation based exemplar selection method. In Section 3.1 and 3.2 we present a geometric interpretation of the exemplar selection model from Section 2.1 and the FFS algorithm from Section 2.2, and study their properties when data is drawn from a union of subspaces. To simplify the analysis, we assume that the self-representation $\boldsymbol{x}_j = \sum_{i \neq j} c_i \boldsymbol{x}_i$ is strictly enforced by extending (4) to $\lambda = \infty$, i.e., we let

$$f_\infty(\boldsymbol{x}_j, \mathcal{X}_0) = \min_{\boldsymbol{c} \in \mathbb{R}^N} \|\boldsymbol{c}\|_1 \ \text{ s.t. } \ \boldsymbol{x}_j = \sum_{i: \boldsymbol{x}_i \in \mathcal{X}_0} c_i \boldsymbol{x}_i. \tag{13}$$

We define $f_\infty(\boldsymbol{x}_j, \mathcal{X}_0) = \infty$ if problem (13) is infeasible. The effect of using a finite λ is discussed in Section 3.3.

3.1 Geometric interpretation

We first provide a geometric interpretation of the exemplars selected by (12). Given any $\mathcal{X}_0$, we denote the convex hull of the symmetrized data points in $\mathcal{X}_0$ by $\mathcal{K}_0$, i.e.,

$$\mathcal{K}_0 := \text{conv}(\pm \mathcal{X}_0) \tag{14}$$

(see an example in Figure 4). The Minkowski functional [55] associated with a set $\mathcal{K}_0$ is given by the following.

Definition 2 (Minkowski functional)**.** The Minkowski functional associated with a set $\mathcal{K}_0 \subseteq \mathbb{R}^D$ is a map denoted by $\|\cdot\|_{\mathcal{K}_0} : \mathbb{R}^D \to \mathbb{R} \cup \{+\infty\}$ and defined by

$$\|\boldsymbol{x}\|_{\mathcal{K}_0} := \inf\{t > 0 : \boldsymbol{x}/t \in \mathcal{K}_0\}. \tag{15}$$

We define $\|\boldsymbol{x}\|_{\mathcal{K}_0} := \infty$ if $\{t > 0 : \boldsymbol{x}/t \in \mathcal{K}_0\}$ is empty.

The Minkowski functional is a norm on span($\mathcal{K}_0$), and its unit ball is $\mathcal{K}_0$. Thus, for any nonzero $\boldsymbol{x} \in$ span($\mathcal{K}_0$), the point $\boldsymbol{x}/\|\boldsymbol{x}\|_{\mathcal{K}_0}$ is the projection onto the boundary of $\mathcal{K}_0$. The green and red dots in Figure 4 are examples of $\boldsymbol{x}$ and $\boldsymbol{x}/\|\boldsymbol{x}\|_{\mathcal{K}_0}$, respectively. It follows that if $\|\boldsymbol{x}\|_2 = 1$, then $1/\|\boldsymbol{x}\|_{\mathcal{K}_0}$ is the length of the ray $\{t\boldsymbol{x} : t \geq 0\}$ inside $\mathcal{K}_0$.

Using Definition 2, it has been shown by [56, Section 2] [18, Section 4.1] that

$$\|\boldsymbol{x}\|_{\mathcal{K}_0} = f_\infty(\boldsymbol{x}, \mathcal{X}_0) \ \text{ for all } \ \boldsymbol{x} \in \mathbb{R}^D. \tag{16}$$

A combination of (16) and the interpretation of $1/\|\boldsymbol{x}\|_{\mathcal{K}_0}$ above provides a geometric interpretation of $f_\infty(\boldsymbol{x}, \mathcal{X}_0)$. That is, $f_\infty(\boldsymbol{x}, \mathcal{X}_0)$ is large if the length of the ray $\{t\boldsymbol{x} : t \geq 0\}$ inside $\mathcal{K}_0$ is small. In particular, it holds that $f_\infty(\boldsymbol{x}, \mathcal{X}_0)$ is infinity if $\boldsymbol{x}$ is not in the span of $\mathcal{X}_0$.

In view of (16), the exemplar selection model (12) may be written equivalently as

$$\mathcal{X}_0^* = \arg\max_{|\mathcal{X}_0| \leq k} \inf_{\boldsymbol{x}_j \in \mathcal{X}} 1/\|\boldsymbol{x}_j\|_{\mathcal{K}_0}. \tag{17}$$

Therefore, the solution to (12) is the subset $\mathcal{X}_0$ of $\mathcal{X}$ that maximizes where the ray $\{t\boldsymbol{x}_j : t \geq 0\}$ intersects $\mathcal{K}_0$ taken over all data $\boldsymbol{x}_j \in \mathcal{X}$ (i.e., maximizes the minimum of such intersections over all $\boldsymbol{x}_j \in \mathcal{X}$).

Also, from (16) we see that each iteration of Algorithm 1 selects the $\boldsymbol{x}_j$ that minimizes $1/\|\boldsymbol{x}_j\|_{\mathcal{K}_0}$. Therefore, each iteration of FFS adds the point $\boldsymbol{x}_j \in \mathcal{X}$ whose associated ray $\{t\boldsymbol{x}_j : t > 0\}$ has the shortest intersection with $\mathcal{K}_0$.

Finally, we remark that our exemplar selection objective is related to the sphere covering problem. This is discussed in detail in the Appendix.

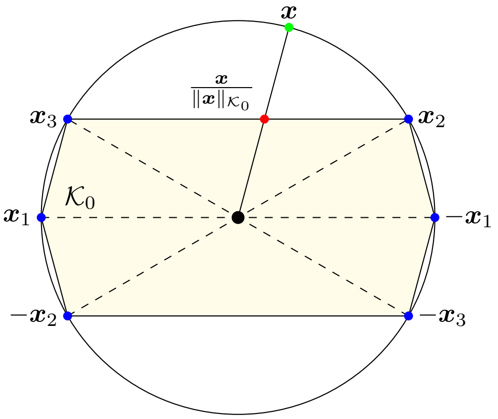


Fig. 4. A geometric illustration of the solution to (12) with $\mathcal{X}_0 = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$. The shaded area is the convex hull $\mathcal{K}_0$ defined in (14).

3.2 Exemplars from a union of subspaces

We now study the properties of our exemplar selection method when applied to data from a union of subspaces. Let $\mathcal{X}$ be drawn from a collection of subspaces $\{\mathcal{S}_\ell\}_{\ell=1}^n$ of dimensions $\{d_\ell\}_{\ell=1}^n$ with each subspace $\mathcal{S}_\ell$ containing at least d_ℓ samples that span $\mathcal{S}_\ell$. We assume that the subspaces are independent, which is commonly used in the analysis of subspace clustering methods [16], [41], [42], [57], [58].

Assumption 1. *The subspaces $\{\mathcal{S}_\ell\}_{\ell=1}^n$ are independent, i.e., $\sum_{\ell=1}^n d_\ell$ is equal to the dimension of $\sum_{\ell=1}^n \mathcal{S}_\ell$.*

We now aim to show that the solution to (12) contains at least d_ℓ independent vectors from each subspace $\mathcal{S}_\ell$ and, moreover, the solution to (2) with $\mathcal{X}_0$ being any solution to (12) is subspace-preserving for all $j \in \{1, \ldots, N\}$. Formally, the subspace-preserving property is defined as follows.

Definition 3 (Subspace-preserving property)**.** A vector $\boldsymbol{c} \in \mathbb{R}^N$ associated with $\boldsymbol{x}_j \in \mathcal{X}$ is called subspace-preserving if $c_i \neq 0$ implies that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are from the same subspace.

We first need the following lemma.

Lemma 4. *Suppose that $\boldsymbol{x}_j \in \mathcal{S}_\ell$. Under Assumption 1, if the optimization problem in (13) is feasible, then any optimal solution $\boldsymbol{c}^*$ to it satisfies $\boldsymbol{x}_j = \sum_{i: \boldsymbol{x}_i \in \mathcal{X}_0 \cap \mathcal{S}_\ell} c_i^* \boldsymbol{x}_i$, and $c_i^* = 0$ for all i satisfying $\boldsymbol{x}_i \notin \mathcal{X}_0 \cap \mathcal{S}_\ell$, i.e., $\boldsymbol{x}_j$ is expressed as a linear combination of points in $\mathcal{X}_0$ that are from its own subspace.*

Proof. An optimal solution $\boldsymbol{c}^*$ to (13) must be feasible, i.e.,

$$\boldsymbol{x}_j = \sum_{i: \boldsymbol{x}_i \in \mathcal{X}_0} c_i^* \boldsymbol{x}_i = \sum_{i: \boldsymbol{x}_i \in \mathcal{X}_0 \cap \mathcal{S}_\ell} c_i^* \boldsymbol{x}_i + \sum_{m \neq \ell} \Big(\sum_{i: \boldsymbol{x}_i \in \mathcal{X}_0 \cap \mathcal{S}_m} c_i^* \boldsymbol{x}_i \Big),$$

which after rearrangement gives

$$\boldsymbol{x}_j - \sum_{i: \boldsymbol{x}_i \in \mathcal{X}_0 \cap \mathcal{S}_\ell} c_i^* \boldsymbol{x}_i = \sum_{m \neq \ell} \Big(\sum_{i: \boldsymbol{x}_i \in \mathcal{X}_0 \cap \mathcal{S}_m} c_i^* \boldsymbol{x}_i \Big). \tag{18}$$

Since the left-hand side is a vector in $\mathcal{S}_\ell$ and the right-hand side is a vector in $\sum_{m \neq \ell} \mathcal{S}_m$, it follows from Assumption 1 and [59, Theorem 6] that $\boldsymbol{x}_j = \sum_{i: \boldsymbol{x}_i \in \mathcal{X}_0 \cap \mathcal{S}_\ell} c_i^* \boldsymbol{x}_i$, as claimed.

Next, let us define the vector $\hat{\boldsymbol{c}}$ such that $\hat{c}_i = c_i^*$ for all $i : \boldsymbol{x}_i \in \mathcal{X}_0 \cap \mathcal{S}_\ell$ and $\hat{c}_i = 0$ for all i such that $\boldsymbol{x}_i \notin \mathcal{X}_0 \cap \mathcal{S}_\ell$.

Using $\boldsymbol{x}_j = \sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0 \cap \mathcal{S}_\ell} c_i^* \boldsymbol{x}_i$ from above and the definition of $\hat{c}$, we see that $\hat{c}$ is feasible for (13). Moreover, it satisfies

$$\|\hat{\boldsymbol{c}}\|_1 = \sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0 \cap \mathcal{S}_\ell} |c_i^*| \leq \|\boldsymbol{c}^*\|_1. \tag{19}$$

Since $\boldsymbol{c}^*$ is optimal for (13), it follows from (19) that $\|\hat{\boldsymbol{c}}\|_1 = \|\boldsymbol{c}^*\|_1$. Combining this fact with (19) shows that $c_i^* = 0$ for all i such that $\boldsymbol{x}_i \notin \mathcal{X}_0 \cap \mathcal{S}_\ell$, which completes the proof. □

We may use this lemma to prove the following result.

Theorem 1. *Under Assumption 1, for all $k \geq \sum_{\ell=1}^n d_\ell$, any solution $\mathcal{X}_0^*$ to the optimization problem* (12) *contains at least d_ℓ linearly independent points from each subspace $\mathcal{S}_\ell$. Moreover, with $\mathcal{X}_0 = \mathcal{X}_0^*$, the optimization problem in* (13) *is feasible for all $\boldsymbol{x}_j \in \mathcal{X}$ with all optimal solutions being subspace-preserving.*

Proof. Let $\mathcal{X}_0^*$ be any optimal solution to (12) for any fixed $k \geq \sum_{\ell=1}^n d_\ell$, and $\mathcal{X}_0 \subseteq \mathcal{X}$ be any subset with $|\mathcal{X}_0| = k$ that contains d_ℓ linearly independent points from $\mathcal{S}_\ell$ for each $\ell \in \{1, \cdots, n\}$, which we know exists. It follows that $f_\infty(\boldsymbol{x}_j, \mathcal{X}_0) < \infty$ for all $\boldsymbol{x}_j \in \mathcal{X}$ so that $F_\infty(\mathcal{X}_0) < \infty$. This and optimality of $\mathcal{X}_0^*$ imply $F_\infty(\mathcal{X}_0^*) \leq F_\infty(\mathcal{X}_0) < \infty$. This fact and the definition of $F_\infty(\mathcal{X}_0^*)$ means that $f_\infty(\boldsymbol{x}_j, \mathcal{X}_0^*) < \infty$ for all j, i.e., $\boldsymbol{x}_j \in \operatorname{span}(\mathcal{X}_0^*)$ for all j. Combining this with Assumption 1 implies that $\mathcal{X}_0^*$ contains at least d_ℓ linearly independent points from each subspace $\mathcal{S}_\ell$, which also means that the problem in (13) is feasible for all $\boldsymbol{x}_j \in \mathcal{X}$. Combining this with Lemma 4 shows that all solutions to the optimization problem in (13) are subspace preserving. □

When $k = \sum_{\ell=1}^n d_\ell$, Theorem 1 shows that d_ℓ points are selected from subspace $\mathcal{S}_\ell$ regardless of the number of points in $\mathcal{S}_\ell$. Therefore, when the data is class imbalanced, (12) selects a subset that is more balanced provided the dimensions of the subspaces do not differ dramatically.

Theorem 1 also shows that only $\sum_{\ell=1}^n d_\ell$ points are needed to correctly represent all data points in $\mathcal{X}$. In other words, the required number of exemplars for representing the dataset does not scale with the size of the dataset $\mathcal{X}$.

Although the FFS algorithm in Section 2.2 is a computationally efficient greedy algorithm that does not necessarily solve (12), the following result shows that it does output a subset of exemplars from the data with desirable properties.

Theorem 2. *The conclusions of Theorem 1 hold when $\mathcal{X}_0^*$ is replaced by $\mathcal{X}_0^{(k)}$ for any $k \geq \sum_{\ell=1}^n d_\ell$, where $\mathcal{X}_0^{(k)}$ is the set of exemplars returned by Algorithm 1 (equivalently, Algorithm 2).*

Proof. Note that since $\lambda = \infty$, it follows from the definition in (13) that $f_\infty(x_j, \mathcal{X}_0^{(i)}) = \infty$ if and only if $x_j \notin \operatorname{span}(\mathcal{X}_0^{(i)})$. It follows from this fact and the construction of Algorithm 1 that each iteration i of Algorithm 1 adds a data point from $\mathcal{X}$ that is linearly independent from those in $\mathcal{X}_0^{(i)}$ provided such a linearly independent vector exists. However, we know from Assumption 1 that there exists $\bar{k} := \sum_{\ell=1}^n d_\ell \leq k$ linearly independent vectors in $\mathcal{X}$. Putting this all together means that $\mathcal{X}_0^{\bar{k}}$ will contain exactly d_ℓ linearly independent points from each subspace $\mathcal{S}_\ell$, which also means that the optimization problem in (13) is feasible for all $\boldsymbol{x}_j \in \mathcal{X}$. Combining this with Lemma 4 completes the proof. □

3.3 Effect of the regularization parameter λ

The analysis in Section 3.1 and 3.2 concentrates on the case where the regularization parameter λ is set to ∞. In real applications where the data $\mathcal{X}$ contains noise and thus deviates from the union-of-subspace model, the self-representation constraint $\boldsymbol{x}_j = \sum_{i \neq j} c_i \boldsymbol{x}_i$ may not be strictly satisfied. In such cases, using a finite λ makes sense.

On the other hand, the value of λ should also not be too small. The following theorem states that if λ is below a certain threshold then the value $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0)$ is the same for all $\boldsymbol{x}_j \in \mathcal{X}$, and therefore no longer provides a measure of how well the data point $\boldsymbol{x}_j$ is represented by $\mathcal{X}_0$. Consequently, Algorithm 1 and Algorithm 2 will fail to produce a useful representative subset of exemplars.

Theorem 3. *Given any $\mathcal{X}_0 \subseteq \mathcal{X}$, we have $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0) = \frac{\lambda}{2}$ for all $\boldsymbol{x}_j \in \mathcal{X} \setminus \mathcal{X}_0$ if*

$$\lambda < \frac{1}{\max\limits_{\boldsymbol{x}' \in \mathcal{X}} \max\limits_{\boldsymbol{x}'' \in \mathcal{X}, \boldsymbol{x}'' \neq \boldsymbol{x}'} |\langle \boldsymbol{x}', \boldsymbol{x}'' \rangle|}. \tag{20}$$

Proof. The optimality condition for the optimization problem in (4) is given by

$$\lambda \boldsymbol{x}_i^\top (\boldsymbol{x}_j - \sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0} c_i \boldsymbol{x}_i) \in \partial |c_i|, \ \ \forall i : \boldsymbol{x}_i \in \mathcal{X}_0, \tag{21}$$

which is satisfied by $\boldsymbol{c} = \boldsymbol{0}$ when (20) is satisfied. Therefore, $\boldsymbol{c} = \boldsymbol{0}$ is an optimal solution to (4). Plugging this solution into the objective function of (4) gives $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0) = \frac{\lambda}{2}$. □

4 The Application of Exemplar Selection to Subspace Clustering and Classification

In this section, we present procedures for using the exemplars returned by Algorithm 2 to generate class assignments for data drawn from a union of subspaces. In Section 4.1 we consider the problem of subspace clustering where the class labels of all data are unknown. In Section 4.2 we consider a setting where the class labels for the exemplars are obtained, and then used to classify the remaining data points.

4.1 Exemplar based subspace clustering

Once a set of exemplars $\mathcal{X}_0$ has been generated, we can compute a representation vector $\boldsymbol{c}_j$ for each $\boldsymbol{x}_j \in \mathcal{X}$ as the solution to the optimization problem (2). As shown in Theorem 2, the vector $\boldsymbol{c}_j$ is expected to be subspace-preserving, i.e., c_{ij} is nonzero only if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are from the same subspace. Motivated by this observation, we use a nearest neighbor approach to compute the segmentation of $\mathcal{X}$ (see Algorithm 3). First, the coefficient vectors $\{\boldsymbol{c}_j\}$ are normalized, i.e., we set $\tilde{\boldsymbol{c}}_j = \boldsymbol{c}_j / \|\boldsymbol{c}_j\|_2$. Then, for each $\tilde{\boldsymbol{c}}_j$ we find t-nearest neighbors with the largest positive inner product with $\tilde{\boldsymbol{c}}_j$. Note that if all vectors in $\{\boldsymbol{c}_j\}_{j=1}^N$ are subspace-preserving, then for any two points $\{\boldsymbol{x}_i, \boldsymbol{x}_j\} \subseteq \mathcal{X}$ we have $\langle \tilde{\boldsymbol{c}}_i, \tilde{\boldsymbol{c}}_j \rangle > 0$ only if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are from the same subspace. Therefore, the t-nearest neighbors of $\tilde{\boldsymbol{c}}_j$ from this step all come from the same subspace as $\boldsymbol{x}_j$. Finally, we compute an affinity matrix from the t-nearest neighbors and apply spectral clustering to get the segmentation[2].

[2] While there could be many other procedures for generating class assignments from the coefficient vectors $\{\boldsymbol{c}_j\}$ such as those in [17], [49], [50], we find in our numerical experiments that the procedure described in Algorithm 3 usually works at least as well.

Algorithm 3 Exemplar based subspace clustering (ESC)

```
Input: Data $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subseteq \mathbb{R}^D$, parameter $\lambda > 1$,
    number of exemplars $k$ and number of neighbors $t$.
1: Compute exemplars $\mathcal{X}_0 = \mathcal{X}_0^{(k)}$ using Algorithm 2. Then
   compute $\{\boldsymbol{c}_j\}_{j=1}^N$ with $\boldsymbol{c}_j$ a solution of (2).
2: Define $\tilde{\boldsymbol{c}}_j = \boldsymbol{c}_j / \|\boldsymbol{c}_j\|_2$ for all $j$. For all $i$ and $j$, set $\mathbf{W}_{ij} =$
   $1$ if $\tilde{\boldsymbol{c}}_j$ is a $t$-nearest neighbor of $\tilde{\boldsymbol{c}}_i$ and $\langle \tilde{\boldsymbol{c}}_j, \tilde{\boldsymbol{c}}_i \rangle > 0$, and
   $\mathbf{W}_{ij} = 0$ otherwise.
3: Set $\mathbf{A} = \mathbf{W} + \mathbf{W}^\top$ and apply spectral clustering to $\mathbf{A}$.
Output: Segmentation of $\mathcal{X}$.
```

Theorem 4. *Take any $k \geq \sum_{\ell=1}^n d_\ell$ and any $t > 0$. Let $\lambda = \infty$. Under Assumption 1, the affinity matrix $\mathbf{A}$ in step 3 of Algorithm 3 has no wrong connections, i.e., the (i, j)-th entry of $\mathbf{A}$ is nonzero only if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are from the same subspace.*

Proof. From Theorem 2 we know that the vectors in $\{\boldsymbol{c}_j\}_{j=1}^N$ computed in step 1 of Algorithm 3 are subspace-preserving. Therefore, $\langle \tilde{\boldsymbol{c}}_j, \tilde{\boldsymbol{c}}_i \rangle > 0$ only if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are from the same subspace. Then, according to the steps for computing $\mathbf{W}$ and $\mathbf{A}$ in Algorithm 3 we know that the (i, j)-th entry of $\mathbf{A}$ is nonzero only if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are from the same subspace. □

By Theorem 4, each nonzero entry in the affinity matrix $\mathbf{A}$ corresponds to pairs of points that are in the same subspace. Although this conclusion holds for all $t > 0$, if the value for t is chosen to be too small, then data points from the same subspace will not form a single connected component in the associated graph, leading to the issue of over-segmentation. In our experiments, we find that t needs to be at least three to produce good clustering performance.

4.2 Exemplar selection for subspace classification

Given a large-scale unlabeled dataset, it is expensive to manually annotate all data. One remedy is to select a small subset of data for manual labeling, and then infer the labels for the remaining data by training a model on the selected subset. In the following, we assume that the exemplars selected by Algorithm 2 have been labeled, and present the sparse representation based classification [60] technique to classify the rest of the data points (see Algorithm 4). For each data point $\boldsymbol{x}_j$, we compute the reconstruction residual with respect to each class ℓ as $\boldsymbol{r}_j^{(\ell)} := \boldsymbol{x}_j - \sum_{i:\boldsymbol{x}_i \in \mathcal{X}_0^{(\ell)}} c_{ij} \boldsymbol{x}_i$, where as above $\boldsymbol{c}_j$ is computed as the solution to (2) for each j, and $\mathcal{X}_0^{(\ell)}$ denotes the subset of the exemplars that are from class ℓ. Note that if $\boldsymbol{c}_j$ is subspace-preserving, then $\boldsymbol{x}_j$ can be represented by exemplars from its own class with zero reconstruction residual. In practice, we expect that $\|\boldsymbol{r}_j^{(\ell)}\|_2 \ll \|\boldsymbol{x}_j\|_2$ for the class ℓ that $\boldsymbol{x}_j$ belongs to, and that $\|\boldsymbol{r}_j^{(\ell)}\|_2 = \|\boldsymbol{x}_j\|_2$ for all other classes. Motivated by this observation, we choose to assign $\boldsymbol{x}_j$ to the class that gives the minimum reconstruction residual.

The following theorem shows that the output of Algorithm 4 is a correct segmentation of the data $\mathcal{X}$.

Theorem 5. *Take any $k \geq \sum_{\ell=1}^n d_\ell$, and let $\lambda = \infty$. Under Assumption 1, the output of Algorithm 4 is such that each point in $\mathcal{X}$ has the correct class label, i.e., the segmentation is correct.*

Algorithm 4 Exemplar selection for subspace classification

```
Input: Data $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subseteq \mathbb{R}^D$, parameter $\lambda > 1$,
    and number of exemplars $k$.
1: Compute exemplars $\mathcal{X}_0 = \mathcal{X}_0^{(k)}$ from Algorithm 2. Then
   compute $\{\boldsymbol{c}_j\}_{j=1}^N$ with $\boldsymbol{c}_j$ a solution to (2).
2: Request the class label of points in $\mathcal{X}_0$. Define $\mathcal{C}_0^{(\ell)} \subseteq \mathcal{X}_0$
   as the subset of exemplars from class $\ell$.
3: Assign each $\boldsymbol{x}_j \in \mathcal{X} \setminus \mathcal{X}_0$ to the class that solves the
   problem $\arg\min_\ell \|\boldsymbol{x}_j - \sum_{i:\boldsymbol{x}_i \in \mathcal{C}_0^{(\ell)}} c_{ij} \boldsymbol{x}_i\|_2$.
Output: Segmentation of $\mathcal{X}$.
```

Proof. Note that in Algorithm 4 we have $\mathcal{X}_0 = \mathcal{X}_0^{(k)}$, where $\mathcal{X}_0^{(k)}$ is the set of exemplars returned by Algorithm 2. Now, consider $\boldsymbol{x}_j \in \mathcal{X} \backslash \mathcal{X}_0$, and assume without loss of generality that $\boldsymbol{x}_j \in \mathcal{S}_\ell$. From Theorem 2, the vector $\boldsymbol{c}_j$ computed in step 1 of Algorithm 4 is subspace-preserving so that

$$\|\boldsymbol{x}_j - \sum_{i:\boldsymbol{x}_i \in \mathcal{C}_0^{(p)}} c_i \boldsymbol{x}_i\|_2 = \|\boldsymbol{x}_j\|_2 = 1 \quad \text{for all } p \neq \ell \tag{22}$$

and

$$\boldsymbol{x}_j - \sum_{i:\boldsymbol{x}_i \in \mathcal{C}_0^{(\ell)}} c_i \boldsymbol{x}_i = \mathbf{0}. \tag{23}$$

From (22), (23), and step 3 of Algorithm 4, it follows that the point $\boldsymbol{x}_j$ is assigned to its correct class, namely ℓ. □

5 Experiments

In this section, we demonstrate the performance of our exemplar selection method for subspace clustering and subspace classification tasks. The sparse optimization problem (2) that must be solved to perform step 7 of Algorithm 2, step 1 of Algorithm 3, and step 1 of Algorithm 4 is solved by the LASSO version of the LARS algorithm [61] implemented in the SPAMS package [62]. The nearest neighbors in step 2 of Algorithm 3 are computed by the k-d tree algorithm implemented in the VLFeat toolbox [63].

Databases. We use three publicly available databases. The Extended MNIST (EMNIST) dataset [64] is an extension of the MNIST dataset that contains gray-scale handwritten digits and letters. We take all 190,998 images corresponding to 26 lower case letters, and use them as the data for a 26-class clustering problem. The size of each image in this dataset is 28 by 28. Following [58], each image is represented by a feature vector computed from a scattering convolutional network [65], which is translational invariant and deformation stable (i.e. it linearizes small deformations). Therefore, these features from EMNIST approximately follow a union of subspaces model.

The German Traffic Sign Recognition Benchmark (GTSRB) database [66] contains 43 categories of street sign data with over 50,000 images in total. We remove categories associated with speed limit and triangle-shaped signs (except the yield sign) as they are difficult to distinguish from each other, which results in a final data set of 12,390 images in 14 categories. Each image is represented by a 1,568-dimensional HOG feature [67] provided with the database. The main intra-class variation in GTSRB is the illumination conditions, therefore the data can be well-approximated by a union of subspaces [6].

For both EMNIST and GTSRB, feature vectors are mean subtracted and projected to dimension 500 by PCA and normalized to have unit ℓ_2 norm. Both the EMNIST and GTSRB databases are imbalanced. In EMNIST, for example, the number of images for each letter ranges from 2,213 (letter "j") to 28,723 (letter "e"), and the number of samples for each letter is approximately equal to their frequencies in the English language. In Figure 1 we show the number of instances for each class in both of these databases.

In order to compare with other methods that are not able to handle large scale datasets, we create a small scale imbalanced dataset from the Extended Yale B face database [68]. The Extended Yale B face database contains images of 38 faces and each of them is taken under 64 different illumination conditions. We randomly select 10 classes and sample a subset from each class. The number of images we sample for those 10 classes is 16 for the first three classes, 32 for the next three classes and 64 for the remaining four classes. The images are preprocessed by standardization (i.e., the images are subtracted by the mean image and divided by the standard deviation) and subsequently normalized to have unit ℓ_2 norm for all methods except for the separable NMF methods as they require nonnegative input.

5.1 Exemplar based subspace clustering

We demonstrate the performance of our exemplar subspace clustering Algorithm 3 (henceforth referred to as ESC-FFS) for subspace clustering on class-imbalanced databases. We set λ to 150, 15 and 100 for EMNIST, GTSRB and Extended Yale B, respectively, and set t to 3 for all three databases.

Baselines. We compare our approach with SSC [17] to show the effectiveness of exemplar selection in addressing imbalanced data. For solving the sparse recovery problem in SSC, we use the algorithm in [69] which is more efficient than the LARS algorithm for large scale problems. For a fair comparison with ESC, we compute an affinity graph for SSC using the same procedure as that used for ESC, i.e., the procedure in Algorithm 3.

We also compare our method with k-means clustering (K-means) and spectral clustering on the k-nearest neighbors graph (Spectral). It is known [70] that Spectral is a provably correct method for subspace clustering. The k-means and k-d trees algorithms used to compute the k-nearest neighbor graph in Spectral are implemented using the VLFeat toolbox [63]. In addition, we compare with the three subspace clustering algorithms SSC-OMP [58], OLRSC [71] and SBC [49], which are able to handle large-scale data. For experiments on the Extended Yale B database, we also include a comparison with LRR [41] and ℓ_0-SSC [44], which cannot effectively handle EMNIST and GTSRB due to memory and running time constraints. For all subspace clustering methods (i.e., SSC-OMP, OLRSC, SBC, LRR and ℓ_0-SSC) we use the code provided by their respective authors.

To demonstrate the advantage of our exemplar selection method, we compare ESC-FFS to an approach we call ESC-Rand, which consists of selecting the exemplars $\mathcal{X}_0$ at random from $\mathcal{X}$, i.e., we replace the exemplar selection via FFS in step 1 of Algorithm 3 by selecting k atoms at random from $\mathcal{X}$ to form $\mathcal{X}_0$. In experiments on Extended Yale B database, we further compare with methods where FFS in step 1 of Algorithm 3 is replaced by other exemplar selection methods including k-centers, K-medoids [72], SMRS [12], kDPP [29], two algorithms for separable NMF (i.e., SPA [37], [73] and Xray [39]), and two algorithms for column subset selection (i.e., GreedyCSS [74] and IPM [35]). For k-centers, we implement the farthest first traversal algorithm (see, e.g. [54]). For K-medoids, we use the function provided by ®Matlab, which employs a variant of the algorithm in [72]. For SMRS, kDPP and GreedyCSS, we use the code provided by their respective authors. For SPA and Xray, we use the code provided by [75]. For IPM, we use our own implementation following the description in [35].

Evaluation metrics. The first metric we use is the clustering accuracy. It measures the maximum proportion of points that are correctly labeled over all possible permutations of the labels. Concretely, let $\{C_1, \cdots, C_n\}$ be the ground-truth partition of the data, $\{G_1, \cdots, G_n\}$ be a clustering result of the same data, $n_{ij} = |C_i \cap G_j|$ be the number of common objects in C_i and G_j, and Π be the set of all permutations of $\{1, \cdots, n\}$. The clustering accuracy is defined as

$$\text{Accuracy} = \max_{\pi \in \Pi} \frac{100}{N} \sum_{i=1}^{n} n_{i,\pi(i)}. \tag{24}$$

In the context of classification, accuracy has been known to be biased when the dataset is class imbalanced [76]. For example, if 99% of a dataset consists of samples from one particular class, then assigning all data points to the same label yields at least 99% accuracy. To address this issue, we also use the F-score averaged over all classes. Let $p_{ij} = n_{ij}/|G_j|$ be the precision and $r_{ij} = n_{ij}/|C_i|$ be the recall. The F-score between the clustering result G_i and the true class C_j is defined as $F_{ij} = \frac{2p_{ij}r_{ij}}{p_{ij}+r_{ij}}$. We report the average F-score given by

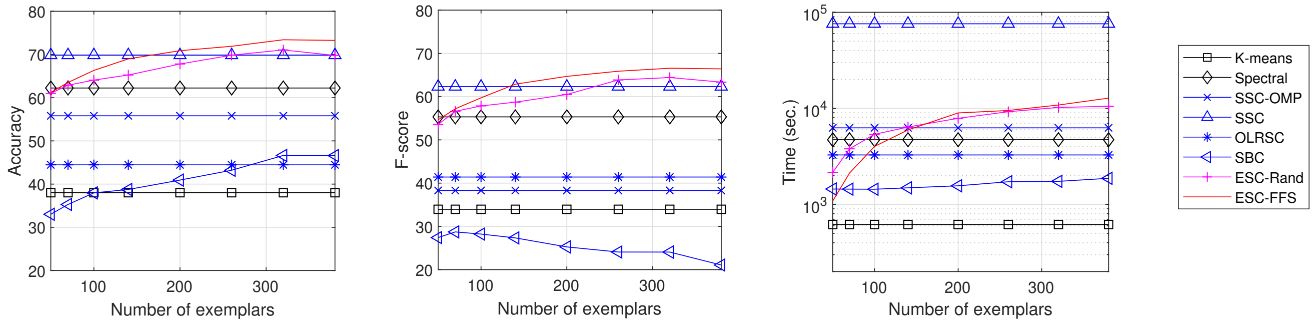
$$\text{F-score} = \max_{\pi \in \Pi} \frac{100}{n} \sum_{i=1}^{n} F_{i,\pi(i)}. \tag{25}$$

Results on EMNIST. Figure 5 shows the results on EMNIST. From left to right, the sub-figures show, respectively, the accuracy, the F-score and the running time (Y axis) as a function of the number of exemplars (X axis). ESC-FFS outperforms all methods except SSC in terms of accuracy and F-score when the number of exemplars is greater than 70.

Recall that in SSC each data point is expressed as a linear combination of all other points. By selecting a subset of exemplars and expressing points using these exemplars, ESC-FFS is able to outperform SSC when the number of exemplars reaches 200. In contrast, ESC-Rand does not outperform SSC by a significant amount, showing the importance of exemplar selection by FFS.

In terms of running time, we see that ESC-FFS is faster than SSC by a large margin. Specifically, ESC-FFS is almost as efficient as ESC-Rand, which indicates that the proposed FFS Algorithm 2 is efficient.

Results on GTSRB. Table 1 reports the clustering performance on the GTSRB database. In addition to reporting average performances, we report the standard deviations. The variation in accuracy and F-score across trials is due to



(a) Accuracy (b) F-score (c) Running time

Fig. 5. Subspace clustering on $190{,}998$ images corresponding to 26 lower case letters from the EMNIST database. We report the averaged accuracy, F-score and running time (in sec.) from 10 trials.

TABLE 1
Subspace Clustering on the GTSRB database. The parameter $k = 160$ is used for ESC-Rand and ESC-FFS. We report the mean and standard deviation for accuracy, F-score and running time (in sec.) from 10 trials.

Methods	Accuracy	F-score	Time (sec.)
K-means	63.7 ± 3.5	54.4 ± 2.8	$\mathbf{12.2} \pm 0.5$
Spectral	89.5 ± 1.3	79.8 ± 2.5	40.3 ± 0.7
SSC-OMP	82.8 ± 0.8	67.8 ± 0.5	22.0 ± 0.2
SSC	92.4 ± 1.1	82.3 ± 2.8	52.2 ± 0.7
OLRSC	71.6 ± 4.3	66.7 ± 4.7	64.9 ± 1.6
SBC	74.9 ± 5.2	72.2 ± 8.5	41.9 ± 0.4
ESC-Rand	89.7 ± 1.6	75.5 ± 4.9	21.5 ± 0.4
ESC-FFS (ours)	$\mathbf{93.0} \pm 1.3$	$\mathbf{85.3} \pm 2.5$	25.2 ± 1.2

1) random initializations of the k-means algorithm, which is used (trivially) in the K-means method, and in the spectral clustering step of all other methods, and 2) random dictionary initialization in OLRSC, SBC, ESC-Rand and ESC-FFS.

We observe that ESC-FFS outperforms all the other methods in terms of accuracy and F-score. In particular, ESC-FFS outperforms SSC, which in turn outperforms ESC-Rand, thus showing the importance of finding a good representative set of exemplars and the effectiveness of FFS in achieving this. In addition, the standard deviation of the accuracy and F-score values for ESC-Rand are larger than for ESC-FFS. This indicates that the set of exemplars given by FFS is more robust in giving reliable clustering results than the randomly selected exemplars in ESC-Rand. In terms of running time, ESC-FFS is also competitive.

Results on Extended Yale B. The averaged accuracy, F-score and running time over 10 randomly sampled imbalanced subsets of the Extended Yale B database are reported in Table 2. Observe that the ℓ_0-SSC and LRR have slightly better performance than SSC, but still are not able to effectively handle imbalanced data. On the other hand, the ESC methods with exemplar selection by kDPP, SMRS, SPA, Xray, GreedyCSS and FFS all have higher accuracies and F-scores than SSC, demonstrating the effectiveness of the exemplar selection approach for handling imbalanced data. In particular, the FFS produces the highest accuracy and the second highest F-score. The k-centers and K-medoids do not demonstrate a significant gain from ESC-Rand. This is because images of the same face lie approximately in a subspace, and their pairwise distances may not be small.

TABLE 2
Subspace Clustering on the Extended Yale B database. The parameter $k = 250$ is used for ESC-FFS. We report the mean and standard deviation for accuracy, F-score and running time (in sec.) from 10 trials.

Methods	Accuracy	F-score	Time (sec.)
K-means	20.1 ± 4.0	18.9 ± 3.9	$\mathbf{1.0} \pm \mathbf{0.2}$
Spectral	46.5 ± 4.5	44.0 ± 5.0	$\mathbf{0.2} \pm \mathbf{1.4}$
SSC-OMP	56.8 ± 4.2	49.9 ± 2.4	0.4 ± 0.03
SSC	67.1 ± 3.6	60.3 ± 4.5	4.6 ± 0.2
OLRSC	30.7 ± 3.1	29.3 ± 3.6	1.9 ± 0.2
SBC	45.4 ± 5.6	43.6 ± 6.2	4.0 ± 0.1
ℓ_0-SSC	67.2 ± 3.6	60.4 ± 4.5	4.6 ± 0.3
LRR	68.3 ± 5.3	61.7 ± 5.4	12.3 ± 0.7
ESC-Rand	65.7 ± 6.3	59.6 ± 8.1	1.4 ± 0.04
ESC-k-centers	67.0 ± 4.0	58.9 ± 3.9	2.0 ± 0.1
ESC-K-medoids	64.5 ± 4.7	56.9 ± 5.6	5.7 ± 0.2
ESC-kDPP	69.7 ± 5.7	$\mathbf{63.0} \pm \mathbf{6.9}$	8.1 ± 0.7
ESC-SMRS	67.9 ± 5.2	60.7 ± 6.1	11.2 ± 2.2
ESC-SPA	67.2 ± 5.2	60.0 ± 4.9	1.8 ± 0.2
ESC-Xray	67.8 ± 5.3	60.5 ± 4.6	231.5 ± 18.5
ESC-GreedyCSS	$\mathbf{70.3} \pm \mathbf{3.3}$	61.4 ± 2.9	1.5 ± 0.1
ESC-IPM	63.4 ± 4.2	56.9 ± 4.4	11.2 ± 2.2
ESC-FFS (ours)	$\mathbf{71.1} \pm \mathbf{4.6}$	$\mathbf{62.4} \pm \mathbf{6.3}$	3.2 ± 0.3

TABLE 3
Effect of varying the parameter t for subspace clustering on Extended Yale B database using ESC-FFS.

t	2	3	4	5	6	8
Accuracy	61.2	70.9	68.3	67.5	65.7	60.4
F-score	54.4	62.1	63.5	62.5	62.5	57.1

Effect of parameter t. In Algorithm 3, segmentation of the data $\mathcal{X}$ from the selected exemplars $\mathcal{X}_0$ is computed by finding the t-nearest neighbors of the representation vector for each data point, then applying spectral clustering to the tNN graph. To understand the role of the parameter t, we conduct additional experiments on the Extended Yale B dataset and report clustering accuracy with varying values of t in Table 3. We can see that both the Accuracy and

TABLE 4
Classification from subsets on the Extended Yale B face database. We report the mean and standard deviation for classification accuracy (%) and running time of the subset selection from 50 trials.

Methods	NN	SRC	SVM	Time (sec.)
Rand	72.0 ± 2.8	85.4 ± 2.1	84.4 ± 2.4	$\mathbf{< 1e-3}$
k-centers	72.8 ± 3.5	86.0 ± 2.4	84.1 ± 2.8	0.2 ± 0.01
K-medoids	78.2 ± 2.7	87.0 ± 1.9	86.7 ± 2.0	1.6 ± 0.1
kDPP	72.8 ± 3.3	88.5 ± 1.8	88.7 ± 2.3	0.4 ± 0.01
SMRS	72.0 ± 2.9	83.8 ± 2.4	82.8 ± 2.7	3.1 ± 0.2
SPA	68.8 ± 4.6	89.1 ± 4.2	90.1 ± 4.3	0.1 ± 0.1
Xray	65.0 ± 6.7	84.3 ± 7.1	83.9 ± 8.4	29.0 ± 2.9
IPM	66.4 ± 3.4	83.4 ± 3.0	83.2 ± 2.8	2.2 ± 0.3
GreedyCSS	$\mathbf{79.4 \pm 2.8}$	$\mathbf{92.1 \pm 1.3}$	91.8 ± 1.7	0.04 ± 0.004
FFS (ours)	70.0 ± 3.4	$\mathbf{92.1} \pm 2.1$	$\mathbf{91.9} \pm 2.5$	0.7 ± 0.1

the F-score are relatively stable for t in the range $[3, 6]$. In addition, we explore an alternative method for computing the segmentation from exemplars. In particular, we replace step 2 and step 3 of Algorithm 3 with the k-means algorithm applied to the set of normalized representation vectors $\{\boldsymbol{c}_j/\|\boldsymbol{c}_j\|_2\}$. This produces an accuracy of 41.6% and an F-score of 40.4%, which are much lower than those obtained with t-nearest neighbors based approaches.

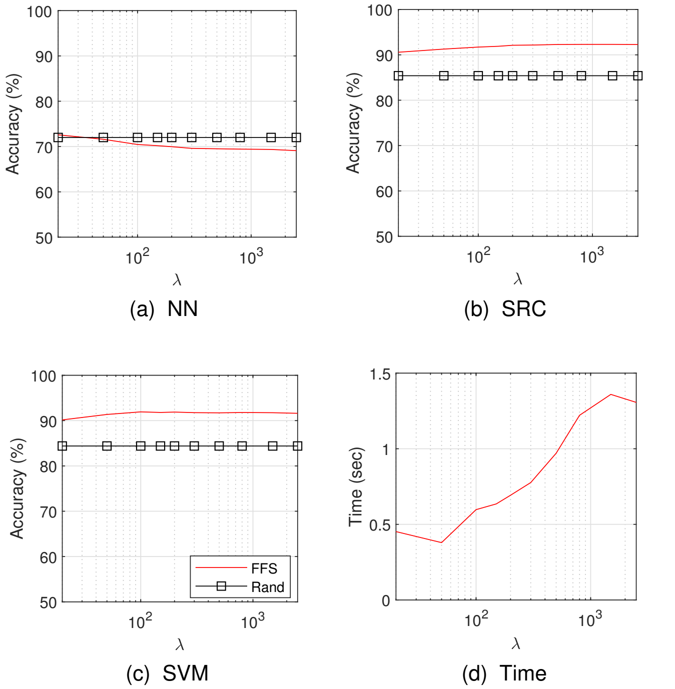
5.2 Exemplar selection for classification

In this section, we evaluate the performance of the FFS algorithm as a tool for selecting a subset of representatives that is subsequently used to classify the entire data set as described in Algorithm 4. The parameter λ in Algorithm 4 is set to 200. The evaluation is performed with the randomly sampled imbalanced subsets of the Extended Yale B database as described in Section 5.1.

In particular, we apply each exemplar selection method to select 100 images from the dataset. Note that during this phase we assume that the ground truth labeling is unknown. After the exemplars have been selected, we assume that the labels of the exemplars are given and use the sparse representation based classification (SRC) method described in Algorithm 4 to assign a label to each of the rest of the data point. In addition to SRC, we also report the results given by the nearest neighbor (NN) and the linear support vector machine (SVM) classifiers.

In Table 4 we report the classification accuracy averaged over 50 trials. We see that the performance with the SRC and SVM classifiers is significantly better than with the NN classifier, which is due to the fact that images of the same face lie approximately in a subspace and their pairwise distance is not necessarily small. In particular, our method obtains the highest accuracy with SRC and SVM.

Effect of parameter λ. To understand the effect of the parameter λ in our FFS algorithm, we conduct experiments with varying values of λ in the range of $[20, 2500]$ and report the classification accuracy and running time in Figure 6. For comparison purposes, we also plot the curves for Rand. It can be seen from Figure 6b and Figure 6c that the accuracy of FFS with SRC and SVM classifiers is non-decreasing as a function of λ in the range of $[20, 200]$ and is mostly insensitive to λ in the range $\lambda > 200$. On the other hand, Figure 6d shows that the running time of FFS increases significantly with λ. This is because the LASSO version of the LARS algorithm that we adopt for solving the optimization problem (4) computes the entire regularization path, therefore a larger value of λ requires more computation.



(a) NN

(b) SRC

(c) SVM

(d) Time

Fig. 6. Effect of varying the parameter λ for classification on Extended Yale B database. The parameter λ is varied along the x-axis from 20 to 2500. Note that the x-axis is in log scale.

Measuring the imbalance. We further evaluate the ability of FFS to handle imbalanced data by measuring the degree of imbalance for the selected representatives. Specifically, since the 64 images in each class of the Extended Yale B database are captured with 64 strobes mounted on a fixed illumination rig, we may assume that all the classes have the same within-class variation and that a best set of exemplars contains equal number of samples from each of the 10 classes. To quantitatively measure the degree of imbalance, we compute the entropy of the proportion of exemplars that

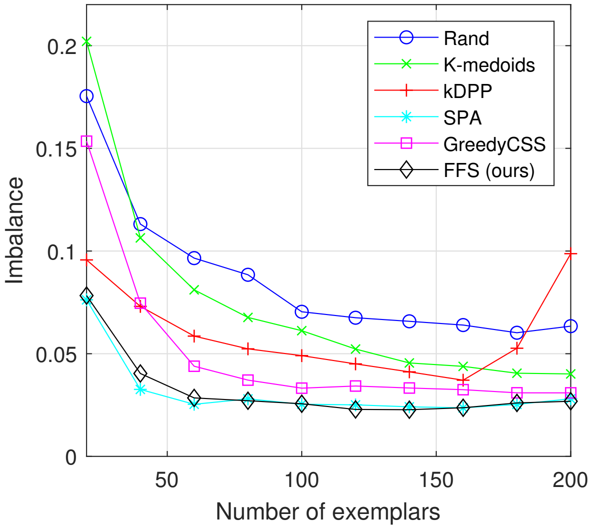


Fig. 7. Performance of exemplar selection for finding a balanced set of representatives from imbalanced classes in the Extended Yale B dataset. We test the methods with the number of representatives varied in the x-axis from 20 to 200, and report the averaged imbalancedness measure from 10 trials.

are selected from each class. That is, we compute

$$\text{Entropy} = -\sum_{i=1}^{10} p_i \log p_i, \quad \text{where } p_i = \frac{s_i}{\sum_{j=1}^{10} s_j}, \tag{26}$$

and s_i is the number of exemplars selected from the i-th class. The entropy is equal to one if and only if $s_1 = s_2 = \cdots = s_{10}$ and is zero if no data point is selected from at least one class. Then, we define the imbalance as one minus the entropy, i.e.,

$$\text{Imbalance} = 1 - \text{Entropy}. \tag{27}$$

Therefore, the imbalance is a non-negative number that takes value 0 if and only if the set of exemplars contains equal number of data points from each class.

The results for FFS, Rand and four other methods that obtain the highest performance in the classification task (i.e., K-medoids, kDPP, SPA and GreedyCSS) are reported in Figure 7. We can see that all methods produce lower imbalance as the number of exemplars increases in the range $[20, 160]$. Note that among the 10 classes in our dataset, each of the 3 smallest classes has only 16 samples. Therefore, it is impossible for the selected set of exemplars to be balanced when the number of such exemplars is greater than 160. This may explain why the imbalance for kDPP, SPA and FFS increases as the number of exemplars goes beyond 160.

In comparing different methods, we observe that random sampling fails to produce a balanced subset because the original data is highly imbalanced. Our FFS significantly outperforms all the other methods for all sizes of subset, with the only exception being SPA, which performs on par with our FFS.

6 Conclusion

We presented a novel approach for unsupervised exemplar selection in a union of subspaces. Our method searches for a set of exemplars from the given dataset such that all data points can be well-represented by the exemplars in terms of a sparse representation cost. When the data comes from a union of subspaces, we proved that our method selects a set of exemplars that is able to represent all data points. We also introduced an algorithm for approximately solving the exemplar selection optimization problem. Empirically, we demonstrated that the exemplars selected by our method can be used for generating a segmentation of the dataset.

Appendix
Relation to the Sphere Covering Problem

We consider the special case when the dataset $\mathcal{X}$ coincides with the unit sphere of $\mathbb{R}^D$, i.e., $\mathcal{X} = \mathbb{S}^{D-1}$. In this case, we establish that our exemplar selection objective in (12) is related to finding k points on the unit sphere with minimal *covering radius*, which is defined in the following.

Definition 4 (Covering radius)**.** The covering radius of a set of points $\mathcal{V} \subseteq \mathbb{S}^{D-1}$ is defined as

$$\gamma(\mathcal{V}) := \max_{\boldsymbol{w} \in \mathbb{S}^{D-1}} \min_{\boldsymbol{v} \in \mathcal{V}} \cos^{-1}(\langle \boldsymbol{v}, \boldsymbol{w} \rangle). \tag{28}$$

The covering radius of the set $\mathcal{V}$ is the minimum angle such that the union of spherical caps centered at each point in $\mathcal{V}$ with this angle covers the entire unit sphere $\mathbb{S}^{D-1}$. Our next result establishes a relationship between the covering radius and our cost function. The proof of the result uses the inradius of a convex body, which is defined as follows.

Definition 5 (inradius)**.** The inradius of a convex set $\mathcal{K}$, denoted by $r(\mathcal{K})$, is the radius of the largest Euclidean ball inscribed in $\mathcal{K}$.

Lemma 5. *For any finite $\mathcal{X}_0 \subseteq \mathcal{X} = \mathbb{S}^{D-1}$, it holds that $F_\infty(\mathcal{X}_0) = 1/\cos\gamma(\pm\mathcal{X}_0)$.*

Proof. From Definition 1, (16), and Definition 2 we have

$$\begin{aligned} F_\infty(\mathcal{X}_0) &= \sup_{\boldsymbol{x} \in \mathbb{S}^{D-1}} f_\infty(\boldsymbol{x}, \mathcal{X}_0) = \sup_{\boldsymbol{x} \in \mathbb{S}^{D-1}} \|\boldsymbol{x}\|_{\mathcal{K}_0} \\ &= \sup_{\boldsymbol{x} \in \mathbb{S}^{D-1}} \inf\{t > 0 : \boldsymbol{x}/t \in \mathcal{K}_0\}. \end{aligned} \tag{29}$$

Then, using Definition 5 and the symmetry of $\mathcal{K}_0$, we have

$$\begin{aligned} r(\mathcal{K}_0) &= \sup\{r > 0 : r\boldsymbol{x} \in \mathcal{K}_0 \text{ for all } \boldsymbol{x} \in \mathbb{S}^{D-1}\} \\ &= \inf_{\boldsymbol{x} \in \mathbb{S}^{D-1}} \sup\{r > 0 : r\boldsymbol{x} \in \mathcal{K}_0\} \\ &= \inf_{\boldsymbol{x} \in \mathbb{S}^{D-1}} \frac{1}{\inf\{t > 0 : \boldsymbol{x}/t \in \mathcal{K}_0\}} \\ &= \frac{1}{\sup_{\boldsymbol{x} \in \mathbb{S}^{D-1}} \inf\{t > 0 : \boldsymbol{x}/t \in \mathcal{K}_0\}}. \end{aligned} \tag{30}$$

By comparing (29) and (30) we have

$$F_\infty(\mathcal{X}_0) = 1/r(\mathcal{K}_0). \tag{31}$$

Furthermore, it was shown in [77, Theorem 9] that $r(\mathcal{K}_0) = \cos\gamma(\pm\mathcal{X}_0)$. Combining this result with (31) allows us to conclude that $F_\infty(\mathcal{X}_0) = 1/\cos\gamma(\pm\mathcal{X}_0)$, as claimed. □

It follows from Lemma 5 that $\arg\min_{|\mathcal{X}_0| \le k} F_\infty(\mathcal{X}_0) = \arg\min_{|\mathcal{X}_0| \le k} \gamma(\pm\mathcal{X}_0)$ when $\mathcal{X} = \mathbb{S}^{D-1}$, i.e., the exemplars $\mathcal{X}_0$ selected by (12) constitute a solution to the problem of finding a subset of $\mathcal{X} = \mathbb{S}^{D-1}$ of size k with minimum covering radius. Note that the covering radius $\gamma(\pm\mathcal{X}_0)$ of the subset $\mathcal{X}_0$ with $|\mathcal{X}_0| \leq k$ is minimized when the points in the symmetrized set $\pm\mathcal{X}_0$ are uniformly distributed on $\mathbb{S}^{D-1}$. The problem of equally distributing points on the sphere without symmetrizing them, i.e. $\min_{|\mathcal{X}_0|\leq k} \gamma(\mathcal{X}_0)$, is known as the sphere covering problem. This problem was first studied by [78] and remains unsolved in geometry [79].

Acknowledgments

C. You, D. P. Robinson and R. Vidal are supported by NSF grant 1618637. C. Li is supported by IARPA grant 127228.

References

[1] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *European conference on machine learning*. Springer, 2004, pp. 39–50.

[2] H. Xiong, J. Wu, and J. Chen, "K-means clustering versus validation measures: a data-distribution perspective," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 318–331, 2009.

[3] X. Chen and D. Cai, "Large scale spectral clustering with landmark-based representation," in *AAAI Conference on Artificial Intelligence*, 2011.

[4] K. Wei, R. Iyer, and J. Bilmes, "Submodularity in data subset selection and active learning," in *International Conference on Machine Learning*, 2015, pp. 1954–1963.

[5] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[6] R. Basri and D. Jacobs, "Lambertian reflection and linear subspaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 218–233, 2003.

[7] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. part i: Greedy pursuit," *Signal Processing*, vol. 86, no. 3, pp. 572–588, 2006.

[8] V. Cevher and A. Krause, "Greedy dictionary selection for sparse representation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 979–988, 2011.

[9] A. Das and D. Kempe, "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection," *arXiv preprint arXiv:1102.3975*, 2011.

[10] J. A. Tropp, "Algorithms for simultaneous sparse approximation. part ii: Convex relaxation," *Signal Processing, Special Issue on Sparse approximations in signal and image processing*, vol. 86, pp. 589–602, 2006.

[11] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, 2011, pp. 3449–3456.

[12] E. Elhamifar, G. Sapiro, and R. Vidal, "See all by looking at a few: Sparse modeling for finding representative objects," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[13] J. Meng, H. Wang, J. Yuan, and Y.-P. Tan, "From keyframes to key objects: Video summarization by representative object proposal selection," in *CVPR*, 2016, pp. 1039–1048.

[14] H. Wang, Y. Kawahara, C. Weng, and J. Yuan, "Representative selection with structured sparsity," *Pattern Recognition*, vol. 63, pp. 268–278, 2017.

[15] Y. Cong, J. Yuan, and J. Luo, "Towards scalable summarization of consumer videos via sparse dictionary selection," *IEEE Transactions on Multimedia*, vol. 14, no. 1, pp. 66–75, 2012.

[16] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2790–2797.

[17] ——, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[18] M. Soltanolkotabi and E. J. Candès, "A geometric analysis of subspace clustering with outliers," *Annals of Statistics*, vol. 40, no. 4, pp. 2195–2238, 2012.

[19] C. You and R. Vidal, "Geometric conditions for subspace-sparse recovery," in *International Conference on Machine Learning*, 2015, pp. 1585–1593.

[20] Y. Wang, Y. Wang, and A. Singh, "A deterministic analysis of noisy sparse subspace clustering for dimensionality-reduced data," in *International Conference on Machine Learning*, 2015, pp. 1422–1431.

[21] Y.-X. Wang and H. Xu, "Noisy sparse subspace clustering," *Journal of Machine Learning Research*, vol. 17, no. 12, pp. 1–41, 2016.

[22] C. You, D. P. Robinson, and R. Vidal, "Provable self-representation based outlier detection in a union of subspaces," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4323–4332.

[23] C.-G. Li, C. You, and R. Vidal, "On geometric analysis of affine sparse subspace clustering," *IEEE Journal on Selected Topics in Signal Processing*, vol. 12, no. 6, pp. 1520–1533, 2018.

[24] D. P. Robinson, R. Vidal, and C. You, "Basis pursuit and orthogonal matching pursuit for subspace-preserving recovery: Theoretical analysis," *arXiv preprint arXiv:1912.13091*, 2019.

[25] C. You, C.-G. Li, D. P. Robinson, and R. Vidal, "Is an affine constraint needed for affine subspace clustering?" in *IEEE International Conference on Computer Vision*, 2019.

[26] C. You, C. Li, D. P. Robinson, and R. Vidal, "A scalable exemplar-based subspace clustering algorithm for class-imbalanced data," in *European Conference on Computer Vision*, 2018.

[27] A. Borodin, "Determinantal point processes," *arXiv preprint arXiv:0911.1153*, 2009.

[28] J. A. Gillenwater, A. Kulesza, E. Fox, and B. Taskar, "Expectation-maximization for learning determinantal point processes," in *NIPS*, 2014, pp. 3149–3157.

[29] A. Kulesza and B. Taskar, "k-dpps: Fixed-size determinantal point processes," in *ICML*, 2011, pp. 1193–1200.

[30] S. Garcia, J. Derrac, J. Cano, and F. Herrera, "Prototype selection for nearest neighbor classification: Taxonomy and empirical study," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 3, pp. 417–435, 2012.

[31] C. Liu, W. Wang, M. Wang, F. Lv, and M. Konan, "An efficient instance selection algorithm to reconstruct training set for support vector machine," *Knowledge-Based Systems*, vol. 116, pp. 58–73, 2017.

[32] T. Chan, "Rank revealing qr factorizations," *Lin. Alg. and its Appl.*, vol. 88-89, pp. 67–82, 1987.

[33] C. Boutsidis, M. W. Mahoney, and P. Drineas, "An improved approximation algorithm for the column subset selection problem," in *Proceedings of SODA*, 2009, pp. 968–977.

[34] J. Altschuler, A. Bhaskara, G. Fu, V. Mirrokni, A. Rostamizadeh, and M. Zadimoghaddam, "Greedy column subset selection: New bounds and distributed algorithms," in *International Conference on Machine Learning*, 2016, pp. 2539–2548.

[35] M. Joneidi, A. Zaeemzadeh, N. Rahnavard, and M. Shah, "Iterative projection and matching: Finding structure-preserving representatives and its application to computer vision," *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[36] M. Joneidi, S. Vahidian, A. Esmaeili, W. Wang, N. Rahnavard, B. Lin, and M. Shah, "Select to better learn: Fast and accurate deep learning using data selection from nonlinear manifolds," 2020.

[37] N. Gillis and S. A. Vavasis, "Fast and robust recursive algorithmsfor separable nonnegative matrix factorization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 4, pp. 698–714, 2013.

[38] S. Arora, R. Ge, R. Kannan, and A. Moitra, "Computing a nonnegative matrix factorization–provably," in *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. ACM, 2012, pp. 145–162.

[39] A. Kumar, V. Sindhwani, and P. Kambadur, "Fast conical hull algorithms for near-separable non-negative matrix factorization," in *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013, pp. 231–239.

[40] E. Elhamifar, G. Sapiro, and R. Vidal, "Finding exemplars from pairwise dissimilarities via simultaneous sparse recovery," in *Neural Information Processing and Systems*, 2012.

[41] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *International Conference on Machine Learning*, 2010, pp. 663–670.

[42] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *European Conference on Computer Vision*, 2012, pp. 347–360.

[43] E. L. Dyer, A. C. Sankaranarayanan, and R. G. Baraniuk, "Greedy feature selection for subspace clustering," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2487–2517, 2013.

[44] Y. Yang, J. Feng, N. Jojic, J. Yang, and T. S. Huang, "ℓ_0-sparse subspace clustering," in *European Conference on Computer Vision*, 2016, pp. 731–747.

[45] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 24–33.

[46] B. Xin, Y. Wang, W. Gao, and D. Wipf, "Building invariances into sparse subspace clustering," *IEEE Transactions on Signal Processing*, vol. 66, no. 2, pp. 449–462, 2018.

[47] Y. Chen, C.-G. Li, and C. You, "Stochastic sparse subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[48] C. You, C. Donnat, D. P. Robinson, and R. Vidal, "A divide-and-conquer framework for large-scale subspace clustering," in *Asilomar Conference on Signals, Systems and Computers*, 2016.

[49] A. Adler, M. Elad, and Y. Hel-Or, "Linear-time subspace clustering via bipartite graph modeling," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2234 – 2246, 2015.

[50] P. A. Traganitis and G. B. Giannakis, "Sketched subspace clustering," *IEEE Transactions on Signal Processing*, 2017.

[51] A. Aldroubi, A. Sekmen, A. B. Koku, and A. F. Cakmak, "Similarity matrix framework for data from union of subspaces," *Applied and Computational Harmonic Analysis*, 2017.

[52] A. Aldroubi, K. Hamm, A. B. Koku, and A. Sekmen, "Cur decompositions, similarity matrices, and subspace clustering," *arXiv preprint arXiv:1711.04178*, 2017.

[53] M. Abdolali, N. Gillis, and M. Rahmati, "Scalable and robust sparse subspace clustering using randomized clustering and multilayer graphs," *arXiv preprint arXiv:1802.07648*, 2018.

[54] D. P. Williamson and D. B. Shmoys, *The design of approximation algorithms*. Cambridge university press, 2011.

[55] R. Vershynin, "Lectures in geometric functional analysis," 2009.

[56] D. L. Donoho, "Neighborly polytopes and sparse solution of underdetermined linear equations," *Technical Report, Stanford University*, 2005.

[57] R. Vidal, R. Tron, and R. Hartley, "Multiframe motion segmentation with missing data using PowerFactorization, and GPCA," *International Journal of Computer Vision*, vol. 79, no. 1, pp. 85–105, 2008.

[58] C. You, D. P. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3918–3927.

[59] K. Hoffman and R. Kunze, "Linear algebra, pr entice-hall," *Inc., Englewood Cliffs, New Jersey*, pp. 122–125, 1971.

[60] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[61] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[62] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *The Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[63] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[64] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "Emnist: an extension of mnist to handwritten letters," *arXiv preprint arXiv:1702.05373*, 2017.

[65] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1872–1886, Aug. 2013. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2012.230

[66] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Networks*, no. 0, pp. –, 2012.

[67] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[68] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[69] C. You, C.-G. Li, D. P. Robinson, and R. Vidal, "Oracle based active set algorithm for scalable elastic net subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3928–3937.

[70] R. Heckel and H. Bölcskei, "Robust subspace clustering via thresholding," *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 6320–6342, 2015.

[71] J. Shen, P. Li, and H. Xu, "Online low-rank subspace clustering by basis dictionary pursuit," in *Proceedings of the 33rd International Conference on Machine Learning*, 2016, pp. 622–631.

[72] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for k-medoids clustering," *Expert systems with applications*, vol. 36, no. 2, pp. 3336–3341, 2009.

[73] H. Ren and C.-I. Chang, "Automatic spectral target recognition in hyperspectral imagery," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1232–1249, 2003.

[74] A. K. Farahat, A. Elgohary, A. Ghodsi, and M. S. Kamel, "Greedy column subset selection for large-scale data sets," *Knowledge and Information Systems*, vol. 45, no. 1, pp. 1–34, 2015.

[75] N. Gillis and S. A. Vavasis, "Semidefinite programming based preconditioning for more robust near-separable nonnegative matrix factorization," *SIAM Journal on Optimization*, vol. 25, no. 1, pp. 677–698, 2015.

[76] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *Pattern recognition (ICPR), 2010 20th international conference on*. IEEE, 2010, pp. 3121–3124.

[77] C. You, "Sparse methods for learning multiple subspaces from large-scale, corrupted and imbalanced data," Ph.D. dissertation, Johns Hopkins University, 2018.

[78] L. F. Toth, "On covering a spherical surface with equal spherical caps (in hungarian)," *Matematikai Fiz. Lapok*, no. 50, pp. 40–46, 1943.

[79] H. T. Croft, R. K. Guy, and K. J. Falconer, *Unsolved problems in geometry*. Springer, 1991.