This ECCV 2018 paper, provided here by the Computer Vision Foundation, is the author-created version.
The content of this paper is identical to the content of the officially published ECCV 2018
LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/eccv

A Scalable Exemplar-based Subspace Clustering Algorithm for Class-Imbalanced Data

Chong You, Chi Li, Daniel P. Robinson, René Vidal

Johns Hopkins University, MD, USA

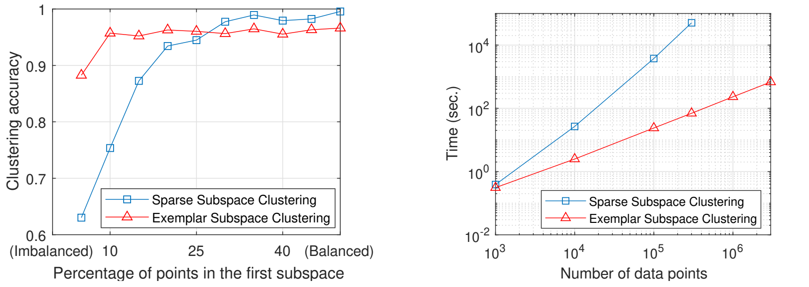
Abstract. Subspace clustering methods based on expressing each data point as a linear combination of a few other data points (e.g., sparse subspace clustering) have become a popular tool for unsupervised learning due to their empirical success and theoretical guarantees. However, their performance can be affected by imbalanced data distributions and large-scale datasets. This paper presents an exemplar-based subspace clustering method to tackle the problem of imbalanced and large-scale datasets. The proposed method searches for a subset of the data that best represents all data points as measured by the ℓ_1 norm of the representation coefficients. To solve our model efficiently, we introduce a farthest first search algorithm which iteratively selects the least well-represented point as an exemplar. When data comes from a union of subspaces, we prove that the computed subset contains enough exemplars from each subspace for expressing all data points even if the data are imbalanced. Our experiments demonstrate that the proposed method outperforms state-of-the-art subspace clustering methods in two large-scale image datasets that are imbalanced. We also demonstrate the effectiveness of our method on unsupervised data subset selection for a face image classification task.

Keywords: Subspace Clustering, Imbalanced Data, Large-scale Data

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 size continues to grow, it has become increasingly 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. Existing labeled databases, such as ImageNet, are manually organized to be class-balanced. On the other hand, the number of data samples in unlabeled datasets varies widely for different classes. Dealing with imbalanced data is hence a major challenge in unsupervised learning tasks.

Traditional unsupervised learning methods exploit the fact that in many computer vision applications the underlying dimension of the data is much smaller than the ambient dimension. For example, it is well-known that the images of a face under varying illumination conditions can be well-approximated by



(a) Clustering on imbalanced data. (b) Clustering on large-scale data.

Fig. 1. Subspace clustering on imbalanced data and large-scale data. (a) x and $100-x$ points (x is varied in the x-axis) are drawn uniformly at random from 2 subspaces of dimension 3 drawn uniformly at random in an ambient space of dimension 5. Note that the clustering accuracy of SSC decreases dramatically as the dataset becomes imbalanced. (b) 10 subspaces of dimension 5 are drawn uniformly at random in an ambient space of dimension 20. An equal number of points is drawn uniformly at random from each subspace. Note that the runtime of SSC increases dramatically with data size.

a 9-dimensional subspace. In practice, computer vision datasets often contain multiple classes, hence they can be modeled by a union of low dimensional subspaces. *Subspace clustering* [1] is a popular approach for unsupervised learning from such data that jointly learns the union of subspaces and assigns each data point to its corresponding subspace.

Many recent subspace clustering methods follow a two-step approach: (1) learn an affinity graph among data points and (2) apply spectral clustering [2] to this graph. In particular, the state-of-the-art methods learn the affinity by exploiting the *self-expressiveness* property [3], 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, Sparse Subspace Clustering (SSC) [3, 4] solves, for each $j \in \{1, 2, \ldots, N\}$, the sparse optimization problem

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

where $\lambda > 0$ and $\boldsymbol{c}_j = [c_{1j}, \cdots, c_{Nj}]^\top$. Subsequently, the affinity between any pair of points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is defined as $|c_{ij}| + |c_{ji}|$. Existing theoretical results for noiseless as well as corrupted data show that, under certain conditions, the solution to (1) is subspace-preserving [4–8], thus justifying the correctness of SSC's affinity. Beyond SSC, many methods have been proposed that use different regularization on the coefficients $\{c_{ij}\}$ [9–14].

Despite the great success of SSC and its variants, previous experimental evaluations focused primarily on balanced datasets, i.e. datasets with an approximately equal 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, as shown in Figure 1(a). Theoretically, we conjecture that the solution to (1) for $\boldsymbol{x}_j$ in an under-represented class is more likely to have nonzero entries corresponding to data points in over-represented classes, which gives false connections in the graph affinity. A proof of this conjecture will be the subject of future work. Another issue with many self-expressiveness based subspace clustering methods is that they are limited to small or medium scale datasets [15]. Figure 1(b) illustrates the running time of SSC as a function of the number of data points N, which is roughly quadratic in N.

Paper contributions. We propose an exemplar-based subspace clustering approach to address the issues of imbalanced and large-scale data. Given a dataset $\mathcal{X}$, the idea is to select a subset $\mathcal{X}_0$, which we call *exemplars*, and write each data point as a linear combination of points in $\mathcal{X}_0$ (rather than $\mathcal{X}$ as in SSC):

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

Observe that (2) is potentially more robust to imbalanced data than (1) in finding subspace-preserving representations when $\mathcal{X}_0$ is balanced across classes. Moreover, (2) can potentially be solved more efficiently than (1) when $\mathcal{X}_0$ is small relative to the original data $\mathcal{X}$. Thus, to achieve robustness to imbalanced data and scalability to large datasets, we need an efficient algorithm for selecting exemplars $\mathcal{X}_0$ that is more balanced across classes.

In this paper, we present a new model for selecting a set of exemplars $\mathcal{X}_0$ that is based on minimizing a maximum representation cost of the data $\mathcal{X}$. (The proofs for results in this paper can be found in [16].) Moreover, we introduce an efficient algorithm for solving the optimization problem that has linear time and memory complexity. Compared to SSC, exemplar-based subspace clustering is less sensitive to imbalanced data and more efficient for big data (see Figure 1). In addition, our work makes the following contributions:

- We present a geometric interpretation of our exemplar selection model and 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.
- We prove that when the data lies in a union of independent subspaces, our method is guaranteed to select sufficiently many data points from each subspace and construct correct data affinities, even when the data is imbalanced.
- We evaluate our method on two imbalanced image datasets: the EMNIST handwritten letter dataset and the GTSRB street sign dataset. Experimental results show that our method outperforms the state-of-the-art in terms of clustering performance and running time.
- We demonstrate through experiments on the Extended Yale B face database that the exemplars selected by our model can be used for unsupervised subset

selection tasks, where the goal is to select a subset from a big data set that may be used to train a classifier that incurs minimum performance loss.

2 Related Work

Sparse dictionary learning (SDL). Sparse representation of a given dataset is a well studied problem in signal processing and machine learning [17, 18]. Given a set $\mathcal{X} \subseteq I\!R^D$ and an integer k, SDL computes a dictionary of atoms $\mathcal{D} \subseteq I\!R^D$ with $|\mathcal{D}| \leq k$ that minimizes the sparse representation cost. Based on SDL, [19] proposed a linear time subspace clustering algorithm that is guaranteed to be correct if the atoms in dictionary $\mathcal{D}$ lie in the same union of subspaces as the input data $\mathcal{X}$. However, there is little evidence that such a condition is satisfied in real data as the atoms of the dictionary $\mathcal{D}$ are not constrained to be a subset of $\mathcal{X}$. Another recent work [20], which used data-independent random matrices as dictionaries, also suffers from this issue and lacks correctness guarantees.

Sparse dictionary selection. Three variations of the SDL model that explicitly constrain the dictionary atoms to be taken from $\mathcal{X}$ are simultaneous sparse representation [21] and dictionary selection [22, 23], which use greedy algorithms to solve their respective optimization problems, and group sparse representative selection [24–29], which uses a convex optimization based approach based on group sparsity. In particular, when the data is drawn from a union of independent subspaces, the method in [26] is shown to select a few representatives from each of the subspaces. However, these methods have quadratic complexity in the number of points in $\mathcal{X}$. Moreover, convex optimization based methods 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.

Subset selection. Selecting a representative subset of the entire data has been studied in a wide range of contexts such as Determinantal Point Processes [30–32], Rank Revealing QR [33], Column subset selection [34, 35], separable Nonnegative Matrix Factorization [36, 37], and so on [38]. However, they do not model data as coming from a union of subspaces and there is no evidence that they can select good representatives from such data. Several recent works [39–41], which use different subset selection methods for subspace clustering, also lack justification that their selected exemplars are representative of the subspaces.

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.

3 Exemplar-based Subspace Clustering (ESC)

In this section, we present our ESC method for clustering a given set of data points $\mathcal{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$. We first formulate the model for selecting a subset $\mathcal{X}_0$ of exemplars from $\mathcal{X}$. Since the model is a combinatorial optimization problem, we present an efficient algorithm for solving it approximately. Finally, we describe the procedure for generating the cluster assignments from the exemplars $\mathcal{X}_0$.

3.1 Exemplar selection via self-representation cost

Without loss of generality, we assume that all data in $\mathcal{X}$ are normalized to have unit ℓ_2 norm. Recall that in SSC, each data point $\boldsymbol{x}_j \in \mathcal{X}$ is written as a linear combination of all other data points with coefficient vector $\boldsymbol{c}_j$. While the nonzero entries in 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 collection of all $\boldsymbol{x}_j$ often needs the whole dataset $\mathcal{X}$. In ESC, the goal is to find a small subset $\mathcal{X}_0 \subseteq \mathcal{X}$ that represents all data points in $\mathcal{X}$. In particular, the set $\mathcal{X}_0$ should contain exemplars from each subspace such that the solution $\boldsymbol{c}_j$ to (2) for each data point $\boldsymbol{x}_j \in \mathcal{X}$ is *subspace-preserving*, i.e. the nonzero entries of $\boldsymbol{c}_j$ correspond to points in the same subspace as $\boldsymbol{x}_j$. In the following, we define a cost function from the optimization in (2) and then present our exemplar selection model.

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

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

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

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

Geometrically, $f_\lambda(\boldsymbol{x}, \mathcal{X}_0)$ measures how well data point $\boldsymbol{x} \in \mathcal{X}$ is covered by the subset $\mathcal{X}_0$ (see Section 4). The function $f_\lambda(\boldsymbol{x}, \mathcal{X}_0)$ has the following properties.

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

Lemma 2. *The value of $f_\lambda(\boldsymbol{x}, \mathcal{X}_0)$ lies in $[1 - \frac{1}{2\lambda}, \frac{\lambda}{2}]$. The lower bound is achieved if and only if $\boldsymbol{x} \in \mathcal{X}_0$ or $-\boldsymbol{x} \in \mathcal{X}_0$, and the upper bound is achieved when $\mathcal{X}_0 = \emptyset$.*

Observe that if $\mathcal{X}_0$ contains enough exemplars from the subspace containing $\boldsymbol{x}_j$ and the optimal solution $\boldsymbol{c}_j$ to (4) is subspace-preserving, then it is expected that $\boldsymbol{c}_j$ will be sparse and that the residual $\boldsymbol{x}_j - X_0\boldsymbol{c}_j$ 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. 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.,

$$\mathcal{X}_0^* = \operatorname*{arg\,min}_{|\mathcal{X}_0| \leq k} F_\lambda(\mathcal{X}_0), \tag{5}$$

where $k \in \mathbb{Z}$ is the target number of exemplars. Note that the objective function $F_\lambda(\cdot)$ in (5) is monotone according to the following result.

Lemma 3. *For any $\emptyset \subseteq \mathcal{X}_0' \subseteq \mathcal{X}_0'' \subseteq \mathcal{X}$, we have $F_\lambda(\mathcal{X}_0') \geq F_\lambda(\mathcal{X}_0'')$.*

Solving the optimization problem (5) 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 the next section, we present an approximate algorithm that is computationally efficient.

3.2 A Farthest First Search (FFS) algorithm for ESC

In Algorithm 1 we present an efficient algorithm for approximately solving (5). The algorithm progressively grows a candidate subset $\mathcal{X}_0$ (initialized as the empty set) until it reaches the desired size k. At each iteration i, step 3 of the algorithm selects the point $\boldsymbol{x} \in \mathcal{X}$ that is worst represented by the current subset $\mathcal{X}_0^{(i)}$ as measured by $f_\lambda(\boldsymbol{x}, \mathcal{X}_0^{(i)})$. A geometric interpretation of this step is presented in Section 4. In particular, it is shown in Lemma 2 that $f_\lambda(\boldsymbol{x}, \mathcal{X}_0^{(i)}) = 1 - \frac{1}{2\lambda}$ for all $\boldsymbol{x} \in \mathcal{X}_0^{(i)}$ and $f_\lambda(\boldsymbol{x}, \mathcal{X}_0^{(i)}) > 1 - \frac{1}{2\lambda}$ if neither $\boldsymbol{x} \in \mathcal{X}_0^{(i)}$ nor $-\boldsymbol{x} \in \mathcal{X}_0^{(i)}$. Thus, $\boldsymbol{x} \notin \mathcal{X}_0^{(i)}$ during every iteration of Algorithm 1.

We also note that the FFS algorithm can be viewed as an extension of the farthest first traversal algorithm (see, e.g. [42]), which is an approximation algorithm for the k-centers problem discussed in Section 2.

Algorithm 1 Farthest first search (FFS) for exemplar selection

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

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

The idea underpinning this computational savings is the monotonicity of $f_\lambda(\boldsymbol{x}, \cdot)$ as discussed in Section 3.1. That is, for any $\emptyset \subseteq \mathcal{X}_0' \subseteq \mathcal{X}_0'' \subseteq \mathcal{X}$ we

Algorithm 2 An efficient implementation of FFS

```
Input: Data $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subseteq I\!R^D$, parameters $\lambda > 1$ and $k$.
 1: Select $\boldsymbol{x} \in \mathcal{X}$ at random and initialize $\mathcal{X}_0^{(1)} \leftarrow \{\boldsymbol{x}\}$.
 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 an ordering 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: $\mathcal{X}_0^{(k)}$
```

have $f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0') \geq f_\lambda(\boldsymbol{x}_j, \mathcal{X}_0'')$. In the FFS algorithm where 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. Using this result, our efficient implementation is outlined in Algorithm 2. 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, our goal is to find a point $\boldsymbol{x} \in \mathcal{X}$ that maximizes $f_\lambda(\boldsymbol{x}, \mathcal{X}_0^{(i)})$. 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 the list $\boldsymbol{x}_{o_1}, \cdots, \boldsymbol{x}_{o_N}$ (step 7) while keeping track of 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 of $f_\lambda(\boldsymbol{x}, \mathcal{X}_0^{(i)})$. 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. Therefore, we can break the loop (step 12) and avoid computing $f_\lambda(\boldsymbol{x}_{o_j}, \mathcal{X}_0^{(i)})$ for the remaining j's.

3.3 Generating cluster assignments from exemplars

After exemplars have been selected by Algorithm 2, we use them to compute a segmentation of $\mathcal{X}$. Specifically, for each $\boldsymbol{x}_j \in \mathcal{X}$ we compute $\boldsymbol{c}_j$ as a solution to the optimization problem (2). As we will see in Theorem 2, the vector $\boldsymbol{c}_j$ is expected to be subspace-preserving. As such, for any two points $\{\boldsymbol{x}_i, \boldsymbol{x}_j\} \subseteq \mathcal{X}$, one has $\langle \boldsymbol{c}_i, \boldsymbol{c}_j \rangle \neq 0$ only if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are from the same subspace.

Algorithm 3 Subspace clustering by ESC-FFS

```
Input: Data $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\} \subseteq I\!\!R^D$, parameters $\lambda > 1$, $k$ and $t$.
Compute $\mathcal{X}_0$ from Algorithm 2, and then compute $\{\boldsymbol{c}_j\}$ from (2). Let $\tilde{\boldsymbol{c}}_j = \boldsymbol{c}_j / \|\boldsymbol{c}_j\|_2$.
Set $W_{ij} = 1$ if $\tilde{\boldsymbol{c}}_j$ is a $t$-nearest neighbor of $\tilde{\boldsymbol{c}}_i$ and 0 otherwise; Set $A = W + W^\top$.
Apply spectral clustering to $A$ to obtain a segmentation of $\mathcal{X}$.
Output: Segmentation of $\mathcal{X}$.
```

Using 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 $\tilde{\boldsymbol{c}}_j$. (Although it is natural to use the t largest inner-products in absolute value, that approach did not perform as well in our numerical experiments.) Finally, we compute an affinity matrix from the t-nearest neighbors and apply spectral clustering to get the segmentation.

4 Theoretical Analysis of ESC

In this section, we present a geometric interpretation of the exemplar selection model from Section 3.1 and the FFS algorithm from Section 3.2, and study their properties in the context of subspace clustering. To simplify the analysis, we assume that the self-representation $\boldsymbol{x}_j = \sum_{i \neq j} c_{ij} \boldsymbol{x}_i$ is strictly enforced by extending (4) to $\lambda = \infty$, i.e., we let

$$f_\infty(\boldsymbol{x}, \mathcal{X}_0) = \min_{\boldsymbol{c} \in \mathbb{R}^N} \|\boldsymbol{c}\|_1 \quad \text{s.t.} \quad \boldsymbol{x} = \sum_{i : \boldsymbol{x}_i \in \mathcal{X}_0} c_{ij} \boldsymbol{x}_i. \tag{6}$$

By convention, we let $f_\infty(\boldsymbol{x}, \mathcal{X}_0) = \infty$ if the optimization problem is infeasible.

4.1 Geometric interpretation

We first provide a geometric interpretation of the exemplars selected by (5). Given any $\mathcal{X}_0$, we denote the convex hull of the symmetrized data points in $\mathcal{X}_0$ as $\mathcal{K}_0$, i.e., $\mathcal{K}_0 := \mathrm{conv}(\pm \mathcal{X}_0)$ (see an example in Figure 2). The Minkowski functional [43] associated with a set $\mathcal{K}_0$ is given by the following.

Definition 2 (Minkowski functional [43]). *The Minkowski functional associated with the set $\mathcal{K}_0 \subseteq I\!\!R^D$ is a map $I\!\!R^D \to R \cup \{+\infty\}$ given by*

$$\|\boldsymbol{x}\|_{\mathcal{K}_0} := \inf\{t > 0 : \boldsymbol{x}/t \in \mathcal{K}_0\}. \tag{7}$$

In particular, we define $\|\boldsymbol{x}\|_{\mathcal{K}_0} := \infty$ if the set $\{t > 0 : \boldsymbol{x}/t \in \mathcal{K}_0\}$ is empty.

Our geometric interpretation is characterized by the reciprocal of $\|\boldsymbol{x}\|_{\mathcal{K}_0}$. The Minkowski functional is a norm in $\mathrm{span}(\mathcal{K}_0)$, the space spanned by $\mathcal{K}_0$, and its unit ball is $\mathcal{K}_0$. Thus, for any $\boldsymbol{x} \in \mathrm{span}(\mathcal{K}_0)$, the point $\boldsymbol{x}/\|\boldsymbol{x}\|_{\mathcal{K}_0}$ is the intersection of the ray $\{t\boldsymbol{x} : t \geq 0\}$ and the boundary of $\mathcal{K}_0$. The green and red dots in Figure 2 are examples of $\boldsymbol{x}$ and $\boldsymbol{x}/\|\boldsymbol{x}\|_{\mathcal{K}_0}$, respectively. It follows that the quantity $1/\|\boldsymbol{x}\|_{\mathcal{K}_0}$ is the length of the ray $\{t\boldsymbol{x} : t \geq 0\}$ inside the convex hull $\mathcal{K}_0$.

Using Definition 2, one can show that the following holds [44, 5]:

$$\|\boldsymbol{x}\|_{\mathcal{K}_0} = f_\infty(\boldsymbol{x}, \mathcal{X}_0) \quad \text{for all} \quad \boldsymbol{x} \in I\!R^D. \tag{8}$$

A combination of (8) 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, $f_\infty(\boldsymbol{x}, \mathcal{X}_0)$ is infinity if $\boldsymbol{x}$ is not in the span of $\mathcal{X}_0$, i.e., $\boldsymbol{x}$ cannot be linearly represented by $\mathcal{X}_0$.

By using (8), the exemplar selection model in (5) is equivalent to computing

$$\mathcal{X}_0^* = \underset{|\mathcal{X}_0| \leq k}{\arg\max} \inf_{\boldsymbol{x} \in \mathcal{X}} 1/\|\boldsymbol{x}\|_{\mathcal{K}_0}. \tag{9}$$

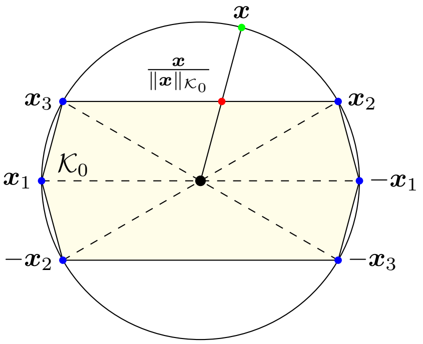


Fig. 2. A geometric illustration of the solution to (5) with $\mathcal{X}_0 = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$. The shaded area is the convex hull $\mathcal{K}_0$.

Therefore, the solution to (5) is the subset $\mathcal{X}_0$ of $\mathcal{X}$ that maximizes the intersection of $\mathcal{K}_0$ and the ray $\{t\boldsymbol{x} : t \geq 0\}$ for every data $\boldsymbol{x} \in \mathcal{X}$ (i.e., maximizes the minimum of such intersections over all $\boldsymbol{x}$).

Furthermore, from (8) we can see that each iteration of Algorithm 1 selects the point $\boldsymbol{x} \in \mathcal{X}$ that minimizes $1/\|\boldsymbol{x}\|_{\mathcal{K}_0}$. Therefore, each iteration of FFS adds the point $\boldsymbol{x}$ that minimizes the intersection of the ray $\{t\boldsymbol{x} : t > 0\}$ with $\mathcal{K}_0$.

Relationship to the sphere covering problem. Let us now consider the special case when the dataset $\mathcal{X}$ coincides with the unit sphere of $I\!R^D$, i.e., $\mathcal{X} = \mathbb{S}^{D-1}$. In this case, we establish that (5) is related to finding the minimum *covering radius*, which is defined in the following.

Definition 3 (Covering radius). *The covering radius of a set of points $\mathcal{V} \subseteq \mathbb{S}^{D-1}$ is defined as*

$$\gamma(\mathcal{V}) := \max_{\boldsymbol{w} \in \mathbb{S}^{D-1}} \min_{\boldsymbol{v} \in \mathcal{V}} \cos^{-1}(\langle \boldsymbol{v}, \boldsymbol{w} \rangle). \tag{10}$$

The covering radius of the set $\mathcal{V}$ can be interpreted as the minimum angle such that the union of spherical caps centered at each point in $\mathcal{V}$ with this radius covers the entire unit sphere $\mathbb{S}^{D-1}$. The following result establishes a relationship between the covering radius and our cost function.

Lemma 4. *For any finite $\mathcal{X}_0 \subseteq \mathcal{X} = \mathbb{S}^{D-1}$ we have $F_\infty(\mathcal{X}_0) = 1/\cos\gamma(\pm\mathcal{X}_0)$.*

It follows from Lemma 4 that $\arg\min_{|\mathcal{X}_0| \leq k} F_\infty(\mathcal{X}_0) = \arg\min_{|\mathcal{X}_0| \leq k} \gamma(\pm\mathcal{X}_0)$ when $\mathcal{X} = \mathbb{S}^{D-1}$, i.e., the exemplars $\mathcal{X}_0$ selected by (5) give the solution to the problem of finding a subset 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 as uniformly distributed on the sphere $\mathbb{S}^{D-1}$ as possible. 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 [45] and remains unsolved in geometry [46].

4.2 ESC on 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 [47, 3, 10, 9, 48].

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 $\sum_{\ell=1}^n \mathcal{S}_\ell$.*

The next result shows that the solution to (5) contains enough exemplars from each subspace.

Theorem 1. *Under Assumption 1, for all $k \geq \sum_{\ell=1}^n d_\ell$, the solution $\mathcal{X}_0^*$ to the optimization problem in* (5) *contains at least d_ℓ linearly independent points from each subspace $\mathcal{S}_\ell$. Moreover, each point $\boldsymbol{x} \in \mathcal{X}$ is expressed as a linear combination of points in $\mathcal{X}_0^*$ that are from its own subspace.*

Theorem 1 shows that when k is set to be $\sum_{\ell=1}^n d_\ell$, then d_ℓ points are selected from subspace $\mathcal{S}_\ell$ regardless of the number of points in that subspace. Therefore, when the data is class imbalanced, (5) is able to select a subset that is more balanced provided that the dimensions of the subspaces do not differ dramatically. This discounts the effect that, when writing a data point as a linear combination of points from $\mathcal{X}$, it is more likely to choose points from oversampled subspaces.

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 3.2 is an approximation algorithm and does not necessarily give the solution to (5), the following result shows that it gives an approximate solution with attractive properties for subspace clustering.

Theorem 2. *The conclusion of Theorem 1 holds for $\mathcal{X}_0^{(k)}$ returned by Algorithm 1 provided $k \geq \sum_{\ell=1}^n d_\ell$.*

Theorem 2 shows that our algorithm FFS is able to select enough samples from each subspace even if the dataset is imbalanced. It also shows that for each data point in $\mathcal{X}$, the representation vector computed in step 1 of Algorithm 3 is subspace-preserving. Formally, we have established the following result.

Theorem 3. *Take any $k \geq \sum_{\ell=1}^n d_\ell$. Under Assumption 1, the representation vectors $\{\boldsymbol{c}_j\}_{j=1}^N$ in step 1 of Algorithm 3 are subspace-preserving, i.e., c_{ij} is nonzero only if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are from the same subspace.*

5 Experiments

In this section, we demonstrate the performance of ESC for subspace clustering as well as for unsupervised subset selection tasks. The sparse optimization problem (4) in step 7 of Algorithm 2 and step 1 of Algorithm 3 are solved by the

LASSO version of the LARS algorithm [49] implemented in the SPAMS package [50]. The nearest neighbors in step 2 of Algorithm 3 are computed by the k-d tree algorithm implemented in the VLFeat toolbox [51].

5.1 Subspace clustering

We first demonstrate the performance of ESC for subspace clustering on large-scale class-imbalanced databases. These databases are described next.

Databases. We use two publicly available databases. The Extended MNIST (EMNIST) dataset [52] 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 [48], each image is represented by a feature vector computed from a scattering convolutional network [53], 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) [54] 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 [55] provided with the database. The major intra-class variation in GTSRB is the illumination conditions, therefore the data can be well-approximated by a union of subspaces [56].

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 3 we show the number of instances for each class in both of these databases.

Baselines. We compare our approach with SSC [4] to show the effectiveness of exemplar selection in addressing imbalanced data. To handle large scale data, we use the efficient algorithm in [12] for solving the sparse recovery problem in SSC. 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 and spectral clustering on the k-nearest neighbors graph, named "Spectral" in the following figures and tables. It is known [57] 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 [51]. In addition, we compare with three other subspace clustering algorithms OMP [48], OLRSC [58] and SBC [19] that are able to handle large-scale data.

We compare these methods with ESC-FFS (Algorithm 3) with λ set to be 150 and 15 for EMNIST and GTSRB, respectively, and t set to be 3 for both

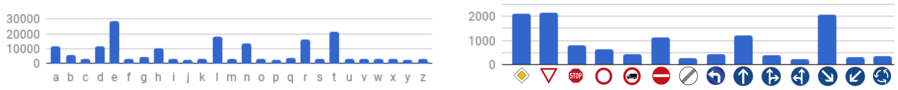


Fig. 3. Number of points in each class of EMNIST (left) and GTSRB (right) databases.

databases. We also report the result of ESC-Rand when the exemplars are selected 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 $\mathcal{X}_0$.

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 $\{1, \cdots, n\}$. Clustering accuracy is defined as

$$\text{Accuracy} = \max_{\pi \in \Pi} \frac{100}{N} \sum_{i=1}^{n} n_{i,\pi(i)}. \tag{11}$$

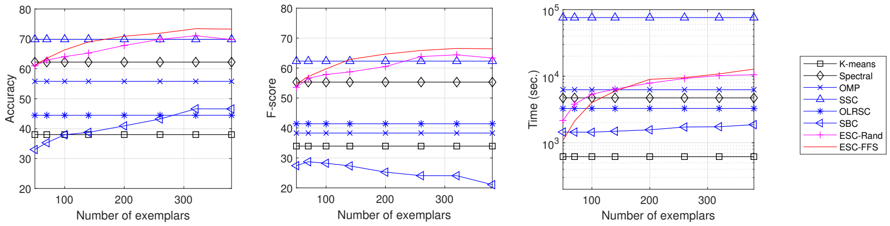
In the context of classification, accuracy has been known to be biased when the dataset is class imbalanced [59]. For example, if a dataset is composed of 99% of samples from one particular class, then assigning all data points to the same label yields at least 99% accuracy. To address this issue, we 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

$$\text{F-score} = \max_{\pi \in \Pi} \frac{100}{n} \sum_{i=1}^{n} F_{i,\pi(i)}. \tag{12}$$

Results on EMNIST. Figure 4 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). We can see that ESC-FFS significantly outperforms all methods except SSC in terms of both 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.



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

Fig. 4. Subspace clustering on images of 26 lower case letters from EMNIST database.

Results on GTSRB. Table 1 reports the clustering performance on the GTSRB database. In addition to reporting average performance, we report the standard deviations. The variation in accuracy and F-score across trials is due to 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 representative set of exemplars and the effectiveness of FFS in achieving this. In addition, the standard deviation of accuracy and F-score for ESC-Rand are all 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.

5.2 Unsupervised subset selection

Given a large-scale unlabeled dataset, it is expensive to manually annotate all data. One solution 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 this section, we evaluate the performance of the FFS algorithm as a tool for selecting a subset of representatives for a given dataset. This subset is then subsequently exploited to classify the entire data set.

We use the Extended Yale B face database, which contains images of 38 faces and each of them is taken under 64 different illumination conditions. For this

Table 1. Subspace clustering on the GTSRB street sign database. The parameter k is fixed to be 160 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.

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

Table 2. 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.

	Rand	k-centers	K-medoids	kDPP	SMRS	FFS
NN	69.4 ± 3.2	69.1 ± 3.7	$\mathbf{75.5} \pm 2.8$	70.5 ± 3.2	69.0 ± 3.1	67.5 ± 4.0
SRC	84.7 ± 2.2	84.9 ± 2.6	86.0 ± 2.1	88.3 ± 2.3	83.4 ± 2.3	$\mathbf{91.4} \pm 2.4$
SVM	83.7 ± 2.5	83.0 ± 2.8	85.3 ± 2.3	87.8 ± 2.1	82.1 ± 2.3	$\mathbf{91.0} \pm 3.0$
Time (sec.)	$< \mathbf{1e-3}$	0.26 ± 0.01	1.5 ± 0.1	0.57 ± 0.06	3.1 ± 0.2	0.70 ± 0.08

experiment, we create an imbalanced dataset by randomly selecting 10 classes and sampling a subset from each class. The number of images we sample for those 10 classes is 16 for the first 3 classes, 32 for the next 3 classes and 64 for the remaining 4 classes. We first apply FFS to select 100 images from this dataset. Note that during this phase we assume that the ground truth labeling is unknown. We then train three classifiers, the nearest neighbor (NN), sparse representation based classification (SRC) [60] and linear support vector machine (SVM) on the selected images, which is then used to classify all of the images.

We compare FFS with random sampling (Rand), k-centers, K-medoids [61], SMRS [26] and kDPP [32]. For k-centers, we implement the farthest first traversal algorithm (see, e.g. [42]). For K-medoids, we use the function provided by ®Matlab, which employs a variant of the algorithm in [61]. For SMRS and kDPP, we use the code provided by the authors. We set $\lambda = 100$ in FFS.

In Table 2 we report the classification accuracy averaged over 50 trials. We can see that the NN classifier works the best with K-medoids, but the performance of NN is worse than SRC and SVM. This is because images of the same face lie approximately in a subspace, and their pairwise distances may not be small. When SRC and SVM are used as classifiers, we can see that our method achieves the best performance.

6 Conclusion

We presented a novel approach to subspace clustering for imbalanced and large-scale data. 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. Analytically, we showed that the set of exemplars selected by our model has the property that its symmetrized convex hull covers as much of the rays $\{t\boldsymbol{x} : t \geq 0\}$ as possible for all data points $\boldsymbol{x} \in \mathcal{X}$. In the context of subspace clustering, we proved that our method selects a set of exemplars that is small and balanced, while being able to represent all data points. We also introduced an algorithm for approximately solving the exemplar selection optimization problem. Empirically we demonstrated that our method is effective for subspace clustering and unsupervised subset selection applications.

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

References

1. Vidal, R.: Subspace clustering. IEEE Signal Processing Magazine **28**(3) (March 2011) 52–68
2. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing **17**(4) (2007) 395–416
3. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: IEEE Conference on Computer Vision and Pattern Recognition. (2009) 2790–2797
4. Elhamifar, E., Vidal, R.: Sparse subspace clustering: Algorithm, theory, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(11) (2013) 2765–2781
5. Soltanolkotabi, M., Candès, E.J.: A geometric analysis of subspace clustering with outliers. Annals of Statistics **40**(4) (2012) 2195–2238
6. You, C., Vidal, R.: Geometric conditions for subspace-sparse recovery. In: International Conference on Machine learning. (2015) 1585–1593
7. Wang, Y.X., Xu, H.: Noisy sparse subspace clustering. Journal of Machine Learning Research **17**(12) (2016) 1–41
8. You, C., Robinson, D., Vidal, R.: Provable self-representation based outlier detection in a union of subspaces. In: IEEE Conference on Computer Vision and Pattern Recognition. (2017)
9. Liu, G., Lin, Z., Yu, Y.: Robust subspace segmentation by low-rank representation. In: International Conference on Machine Learning. (2010) 663–670
10. Lu, C.Y., Min, H., Zhao, Z.Q., Zhu, L., Huang, D.S., Yan, S.: Robust and efficient subspace segmentation via least squares regression. In: European Conference on Computer Vision. (2012) 347–360
11. Dyer, E.L., Sankaranarayanan, A.C., Baraniuk, R.G.: Greedy feature selection for subspace clustering. Journal of Machine Learning Research **14**(1) (2013) 2487–2517
12. You, C., Li, C.G., Robinson, D., Vidal, R.: Oracle based active set algorithm for scalable elastic net subspace clustering. In: IEEE Conference on Computer Vision and Pattern Recognition. (2016) 3928–3937
13. Yang, Y., Feng, J., Jojic, N., Yang, J., Huang, T.S.: ℓ_0-sparse subspace clustering. In: European Conference on Computer Vision. (2016) 731–747
14. Xin, B., Wang, Y., Gao, W., Wipf, D.: Building invariances into sparse subspace clustering. IEEE Transactions on Signal Processing **66**(2) (2018) 449–462
15. You, C., Donnat, C., Robinson, D., Vidal, R.: A divide-and-conquer framework for large-scale subspace clustering. In: Asilomar Conference on Signals, Systems and Computers. (2016)
16. You, C., Li, C., Robinson, D., Vidal, R.: A scalable exemplar-based subspace clustering algorithm for class-imbalanced data. (2018)
17. Aharon, M., Elad, M., Bruckstein, A.M.: K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing **54**(11) (2006) 4311–4322
18. Bach, F., Jenatton, R., Mairal, J., Obozinski, G.: Optimization with sparsity-inducing penalties. Journal Foundations and Trends in Machine Learning **4**(1) (2012) 1–106
19. Adler, A., Elad, M., Hel-Or, Y.: Linear-time subspace clustering via bipartite graph modeling. IEEE Transactions on Neural Networks and Learning Systems **26**(10) (2015) 2234 – 2246
20. Traganitis, P.A., Giannakis, G.B.: Sketched subspace clustering. IEEE Transactions on Signal Processing (2017)

21. Tropp, J.A., Gilbert, A.C., Strauss, M.J.: Algorithms for simultaneous sparse approximation. part i: Greedy pursuit. Signal Processing **86**(3) (2006) 572–588
22. Cevher, V., Krause, A.: Greedy dictionary selection for sparse representation. IEEE Journal of Selected Topics in Signal Processing **5**(5) (2011) 979–988
23. Das, A., Kempe, D.: Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. arXiv preprint arXiv:1102.3975 (2011)
24. Tropp, J.A.: Algorithms for simultaneous sparse approximation. part ii: Convex relaxation. Signal Processing, Special Issue on Sparse approximations in signal and image processing **86** (2006) 589–602
25. Cong, Y., Yuan, J., Liu, J.: 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) 3449–3456
26. Elhamifar, E., Sapiro, G., Vidal, R.: See all by looking at a few: Sparse modeling for finding representative objects. In: IEEE Conference on Computer Vision and Pattern Recognition. (2012)
27. Meng, J., Wang, H., Yuan, J., Tan, Y.P.: From keyframes to key objects: Video summarization by representative object proposal selection. In: CVPR. (2016) 1039–1048
28. Wang, H., Kawahara, Y., Weng, C., Yuan, J.: Representative selection with structured sparsity. Pattern Recognition **63** (2017) 268–278
29. Cong, Y., Yuan, J., Luo, J.: Towards scalable summarization of consumer videos via sparse dictionary selection. IEEE Transactions on Multimedia **14**(1) (2012) 66–75
30. Borodin, A.: Determinantal point processes. arXiv preprint arXiv:0911.1153 (2009)
31. Gillenwater, J.A., Kulesza, A., Fox, E., Taskar, B.: Expectation-maximization for learning determinantal point processes. In: NIPS. (2014) 3149–3157
32. Kulesza, A., Taskar, B.: k-dpps: Fixed-size determinantal point processes. In: ICML. (2011) 1193–1200
33. Chan, T.: Rank revealing qr factorizations. Lin. Alg. and its Appl. **88-89** (1987) 67–82
34. Boutsidis, C., Mahoney, M.W., Drineas, P.: An improved approximation algorithm for the column subset selection problem. In: Proceedings of SODA. (2009) 968–977
35. Altschuler, J., Bhaskara, A., Fu, G., Mirrokni, V., Rostamizadeh, A., Zadimoghaddam, M.: Greedy column subset selection: New bounds and distributed algorithms. In: International Conference on Machine Learning. (2016) 2539–2548
36. Arora, S., Ge, R., Kannan, R., Moitra, A.: Computing a nonnegative matrix factorization–provably. In: Proceedings of the forty-fourth annual ACM symposium on Theory of computing, ACM (2012) 145–162
37. Kumar, A., Sindhwani, V., Kambadur, P.: Fast conical hull algorithms for near-separable non-negative matrix factorization. In: Proceedings of the 30th International Conference on Machine Learning (ICML). (2013) 231–239
38. Elhamifar, E., Sapiro, G., Vidal, R.: Finding exemplars from pairwise dissimilarities via simultaneous sparse recovery. In: Neural Information Processing and Systems. (2012)
39. Aldroubi, A., Sekmen, A., Koku, A.B., Cakmak, A.F.: Similarity matrix framework for data from union of subspaces. Applied and Computational Harmonic Analysis (2017)
40. Aldroubi, A., Hamm, K., Koku, A.B., Sekmen, A.: Cur decompositions, similarity matrices, and subspace clustering. arXiv preprint arXiv:1711.04178 (2017)

41. Abdolali, M., Gillis, N., Rahmati, M.: Scalable and robust sparse subspace clustering using randomized clustering and multilayer graphs. arXiv preprint arXiv:1802.07648 (2018)
42. Williamson, D.P., Shmoys, D.B.: The design of approximation algorithms. Cambridge university press (2011)
43. Vershynin, R.: Lectures in geometric functional analysis. (2009)
44. Donoho, D.L.: Neighborly polytopes and sparse solution of underdetermined linear equations. Technical Report, Stanford University (2005)
45. Toth, L.F.: On covering a spherical surface with equal spherical caps (in hungarian). Matematikai Fiz. Lapok (50) (1943) 40–46
46. Croft, H.T., Guy, R.K., Falconer, K.J.: Unsolved problems in geometry. Springer (1991)
47. Vidal, R., Tron, R., Hartley, R.: Multiframe motion segmentation with missing data using PowerFactorization, and GPCA. International Journal of Computer Vision **79**(1) (2008) 85–105
48. You, C., Robinson, D., Vidal, R.: Scalable sparse subspace clustering by orthogonal matching pursuit. In: IEEE Conference on Computer Vision and Pattern Recognition. (2016) 3918–3927
49. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. Annals of Statistics **32**(2) (2004) 407–499
50. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. The Journal of Machine Learning Research **11** (2010) 19–60
51. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/ (2008)
52. Cohen, G., Afshar, S., Tapson, J., van Schaik, A.: Emnist: an extension of mnist to handwritten letters. arXiv preprint arXiv:1702.05373 (2017)
53. Bruna, J., Mallat, S.: Invariant scattering convolution networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(8) (2013) 1872–1886
54. Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. Neural Networks (0) (2012) –
55. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition. (2005)
56. Basri, R., Jacobs, D.: Lambertian reflection and linear subspaces. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(2) (2003) 218–233
57. Heckel, R., Bölcskei, H.: Robust subspace clustering via thresholding. IEEE Transactions on Information Theory **61**(11) (2015) 6320–6342
58. Shen, J., Li, P., Xu, H.: Online low-rank subspace clustering by basis dictionary pursuit. In: Proceedings of the 33rd International Conference on Machine Learning. (2016) 622–631
59. Brodersen, K.H., Ong, C.S., Stephan, K.E., Buhmann, J.M.: The balanced accuracy and its posterior distribution. In: Pattern recognition (ICPR), 2010 20th international conference on, IEEE (2010) 3121–3124
60. Wright, S.J., Nowak, R.D., Figueiredo, M.A.T.: Sparse reconstruction by separable approximation. IEEE Transactions on Signal Processing **57** (2009) 2479–2493
61. Park, H.S., Jun, C.H.: A simple and fast algorithm for k-medoids clustering. Expert systems with applications **36**(2) (2009) 3336–3341