Select to *Better* Learn: Fast and Accurate Deep Learning using Data Selection from Nonlinear Manifolds

Mohsen Joneidi*, Saeed Vahidian†, Ashkan Esmaeili*, Weijia Wang†, Nazanin Rahnavard*, Bill Lin†, and Mubarak Shah#

* University of Central Florida, Department of Electrical and Computer Engineering

† University of California, San Diego, Department of Electrical and Computer Engineering

University of Central Florida, Center for Research in Computer Vision

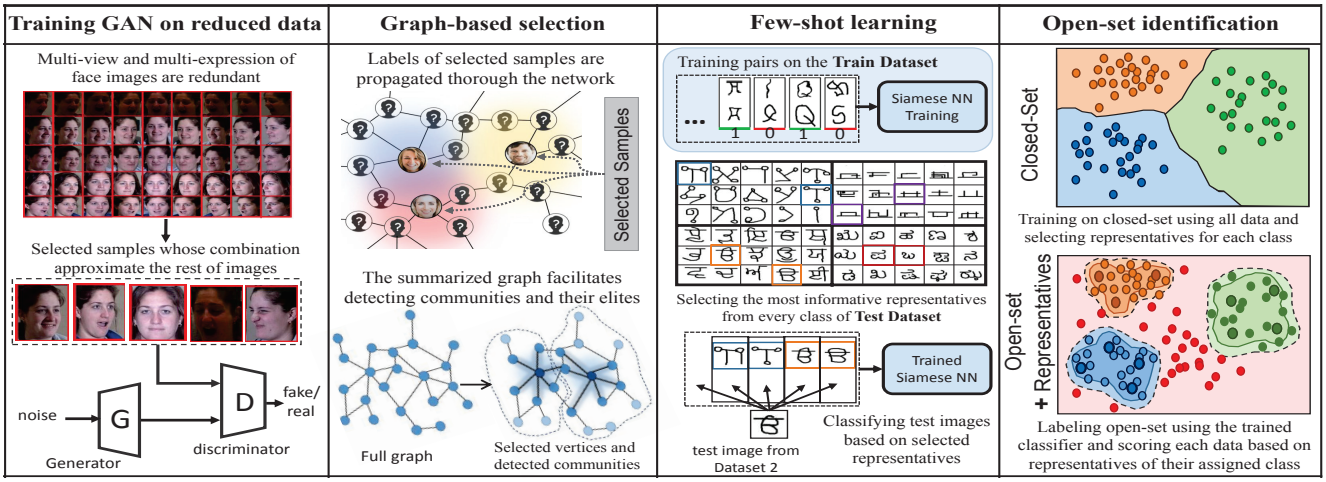

Figure 1: Several deep learning applications of our proposed data selection algorithms discussed in this paper.

Abstract

Finding a small subset of data whose linear combination spans other data points, also called column subset selection problem (CSSP), is an important open problem in computer science with many applications in computer vision and deep learning such as the ones shown in Fig. 1. There are some studies that solve CSSP in a polynomial *time complexity w.r.t. the size of the original dataset. A simple and efficient selection algorithm with a linear complexity order, referred to as* spectrum pursuit *(SP), is proposed that pursuits spectral components of the dataset using available sample points. The proposed non-greedy algorithm aims to iteratively find K data samples whose span is close to that of the first K spectral components of entire data. SP has no parameter to be fine tuned and this desirable property makes it problem-independent. The simplicity of SP enables us to extend the underlying linear model to more complex models such as nonlinear manifolds and graph-based models. The nonlinear extension of SP is introduced as* kernel-SP *(KSP). The superiority of the proposed algorithms is demonstrated in a wide range of applications.*

1. Introduction

Processing M data samples, each including N features, is not feasible for of most the systems, when M is a very large number. Therefore, it is crucial to select a small subset of $K << M$ data from the entire set such that the selected data can capture the underlying properties or structure of the entire data. This way, complex systems such as deep learning (DL) networks can operate on the informative selected data rather than the redundant entire data. Randomly selecting K out of M data, while computationally simple, is inefficient in many cases, since non-informative or redundant instances may be among the selected ones. On the other hand, the optimal selection of data for a specific task involves solving an NP-hard problem [2]. For example, finding an optimal subset of data to be employed in training a DL network, with the best performance, requires $\binom{M}{K}$ number of trial and errors, which is not tractable. It is essential to define a versatile objective function and to develop a method that efficiently selects the K samples that optimize the function. Let us assume the M data samples are organized as the columns of a matrix $\boldsymbol{A} \in \mathbb{R}^{N \times M}$. The following is a general purpose cost function for subset se-

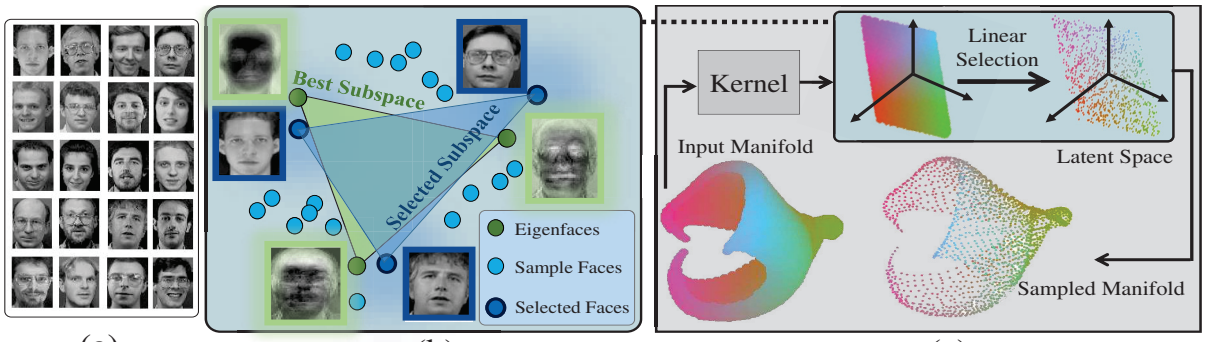

Figure 2: Intuitive illustration of our main contributions in this paper. (a) A dataset including 20 real images from AT&T face database [1] is considered. (b) the images in (a) are represented as blue dots. Three most significant eigenfaces are shown by green dots. However, these eigenfaces are not among data samples. Here, we are interested in selecting the best 3 out 20 real images, whose span is the closest to the span of the 3 eigenfaces. There are $\binom{20}{3}$ possible combinations from which the best subset must be selected. In this paper, we propose the SP algorithm to select K samples such that their span pursuits the span of the first K singular vectors. (c) Utilizing the proposed linear selection algorithm (SP), a tractable algorithm is developed for selecting from low-dimensional manifolds. First, a kernel which is defined by neighborhood transforms the given data on a manifold to a latent space. Next, the linear selection is performed.

lection, known as *column subset selection problem* (CSSP) [3]:

$$\mathbb{S}^* = \underset{|\mathbb{S}|\leq K}{\operatorname{argmin}} \|\boldsymbol{A} - \pi_{\mathbb{S}}(\boldsymbol{A})\|_F^2, \tag{1}$$

where $\pi_{\mathbb{S}}$ is the linear projection operator on the span of K columns of $\boldsymbol{A}$ indicated by set $\mathbb{S}$. This is an open problem which has been shown to be NP hard [4, 2]. Moreover, the cost function is not sub-modular [5], and greedy algorithms are not efficient to tackle Problem (1). Computer scientists and mathematicians during the last 30 years have proposed many tractable selection algorithms that guarantee an upper bound for the projection error $\|\boldsymbol{A}-\pi_{\mathbb{S}}(\boldsymbol{A})\|_F^2$. These works include algorithms based on QR decomposition of matrix $\boldsymbol{A}$ with column pivoting (QRCP) [6, 7, 8], methods based on volume sampling (VS) [9, 10, 11] and matrix subset selection algorithms [3, 12, 13]. However, the guaranteed upper bounds are very loose and the corresponding selection results are far from the actual minimizer of CSSP in practice. Interested readers are referred to [14, 12] and Sec. 2.1 in [15] for detailed discussions. For example, in VS it is shown that the projection error on the span of K selected samples is guaranteed to be less than $K+1$ times of the projection error on the span of the K first left singular vectors (which is too loose for a large K). Recently, it was shown that VS performs even worse than random selection in some scenarios [16]. Moreover, some efforts have been made using convex relaxation and regularization. Fine tuning of these methods is not straightforward. Moreover their cubical complexity is an obstacle to employ these methods for diverse applications.

Recently, a low-complexity approach was proposed to solve CSSP, referred to as iterative projection and matching (IPM) [17]. IPM is a greedy algorithm that selects K consecutive and locally optimum samples, without the option of revisiting the previous selections and escaping local optima. Moreover, IPM samples the data from *linear subspaces*, while in general data points reside in the union of *nonlinear manifolds*.

In this paper, an efficient non-greedy algorithm is proposed to solve Problem (1) with a linear order of complexity. The proposed subspace-based algorithm outperforms the state-of-the-art algorithms in terms of accuracy for CSSP. In addition, the simplicity and accuracy of the proposed algorithm enable us to extend it for efficient sampling from nonlinear manifolds. The intuition behind our work is depicted in Fig. 2. Assume for solving CSSP, we are not restricted to selecting representatives from data samples, and we are allowed to generate pseudo-data and select them as representatives. In this scenario, the best K representatives are the first K spectral components of data according to definition of singular value decomposition (SVD) [18]. However, the spectral components are not among the data samples. Our proposed algorithm aims to find K data samples such that their span is close to that of the first K spectrum of data. We refer to our proposed algorithm as *spectrum pursuit (SP)*. Fig. 2 (b) shows the intuition behind SP and Fig. 2 (c) shows a straightforward extension of SP for sampling from nonlinear manifolds. We refer to this algorithm as *Kernel Spectrum Pursuit (KSP)*.

Our main contributions can be summarized as:

- We introduce SP, a non-greedy selection algorithm with linear order complexity w.r.t. the number of original data points. SP captures spectral characteristics of dataset using only a small number of samples. To the best of our knowledge, SP is the most accurate solver for CSSP.
- Further, we extend SP to Kernel-SP for manifold-based data selection.

- We provide extensive evaluations to validate our proposed selection schemes. In particular, we evaluate the proposed algorithms on training generative adversarial networks, graph-based label propagation, few shot classification, and open-set identification, as shown in Fig. 1. We demonstrate that our proposed algorithms outperform the state-of-the-art algorithms.

2. Data Selection from Linear Subspaces

In this section, we first introduce the related work on matrix subset selection and then we propose our algorithm for CSSP.

2.1. Related Work

A simple approach to selection is to reduce the entire data and evaluate a criterion *only* for the reduced set, $\boldsymbol{A}_{\mathbb{S}}$. Mathematically speaking, we need to solve the following problem [19, 10]:

$$\mathbb{S}^* = \underset{|\mathbb{S}|\leq K}{\operatorname{argmin}} \ \phi\left((\boldsymbol{A}_{\mathbb{S}}^T \boldsymbol{A}_{\mathbb{S}})^{-1}\right). \tag{2}$$

Here, $\phi(.)$ is a function of matrix eigenvalues, such as the determinant or trace function. This is an NP hard and non-convex problem that can be solved via convex relaxation of ℓ_0 norm with time complexity of $O(M^3)$ [20, 19]. There are several other efforts in this area for designing function ϕ [10, 21, 22, 23]. Inspired by D-optimal design, VS [11] considers a selection probability for each subset of data, which is proportional to the determinant (volume) of the reduced matrix [10, 24, 25]. To the best of our knowledge the tightest bound for selecting K columns [26] for CSSP, introduced in a paper published in NIPS 2019,is as follows:

$$\|\boldsymbol{A} - \pi_{\mathbb{S}^*}(\boldsymbol{A})\|_F^2 \leq \sqrt{K+1}\, \|\boldsymbol{A} - \boldsymbol{A}_K\|_F^2,$$

where $\boldsymbol{A}_K$ is the best rank-K approximation of $\boldsymbol{A}$. Moreover, VS guarantees a projection error up to $K+1$ times worse than the first K singular vectors [11]. A set of diverse samples optimizes cost function (2) and algorithms such as VS assign a higher probability for them to be chosen. However, selecting some diverse samples that are solely different from each other probably does not provide good representative for all (un-selected) data.

Ensuring that selected samples are able to reconstruct un-selected samples is a more robust approach than selecting a diverse subset. The exact solution of Problem (1) aims to find such a subset. An equivalent problem to the original problem (1) is proposed in [27]. Their suggested equivalent problem exploits the mixed norm, $\|.\|_{2,0}$, which is not a convex function and they propose to employ ℓ_1 regularization to relax it [27]. There is no guarantee that convex relaxation provides the best approximation for an NP-hard problem. Furthermore, such methods which approach the problem using convex programming are usually computationally intensive for large datasets [27, 28, 29, 30]. In this paper, we present another reformulation of Problem (1) and propose a fast and accurate algorithm for addressing CSSP.

2.2. Spectrum Pursuit (SP)

Projection of all data onto the subspace spanned by K columns of $\boldsymbol{A}$, indexed by $\mathbb{S}$, i.e., $\pi_{\mathbb{S}}(\boldsymbol{A})$, can be expressed by a rank-K factorization, $\boldsymbol{U}\boldsymbol{V}^T$. In this factorization, $\boldsymbol{U} \in \mathbb{R}^{N\times K}$, $\boldsymbol{V} \in \mathbb{R}^{M\times K}$, and $\boldsymbol{U}$ includes a set of K normalized columns of $\boldsymbol{A}$, indexed by $\mathbb{S}$. Therefore, the optimization problem (1) can be restated as [17]:

$$\underset{\boldsymbol{U},\boldsymbol{V}}{\operatorname{argmin}} \ \|\boldsymbol{A} - \boldsymbol{U}\boldsymbol{V}^T\|_F^2; \ \text{s.t. } \boldsymbol{u}_k \in \mathbb{A}, \tag{3}$$

where $\mathbb{A} = \{\tilde{\boldsymbol{a}_1}, \tilde{\boldsymbol{a}_2}, \ldots, \tilde{\boldsymbol{a}}_M\}$, $\tilde{\boldsymbol{a}}_m = \boldsymbol{a}_m/\|\boldsymbol{a}_m\|_2$, and $\boldsymbol{u}_k$ is the k^{th} column of $\boldsymbol{U}$. It should be noted that $\boldsymbol{U}$ is restricted to be a collection of K normalized columns of $\boldsymbol{A}$, while there is no constraint on $\boldsymbol{V}$. As mentioned before, this is an NP hard problem. Recently, IPM [17], a fast sub-optimal and greedy approach to tackle (3), was proposed. In IPM, samples are iteratively selected in a greedy manner until K samples are collected. In this paper, we propose a new selection algorithm, referred to as Spectrum Pursuit (SP), which can select a more accurate solution for Problem (3) than that of IPM. The time complexity of both IPM and SP are linear with respect to the number of samples and the dimension of samples, which is desirable for selection from very large datasets. Our proposed SP algorithm facilitates *revising* our selection in each iteration and escaping from local optima. In SP, we modify (3) into two sub-problems. The first one is built upon the assumption that we have already selected $K-1$ data points and the goal is to select the next best data. However, it relaxes the constraint $\boldsymbol{u}_k \in \mathbb{A}$ in (3) to a moderate constraint $\|\boldsymbol{u_k}\| = 1$. This relaxation makes finding the solution tractable at the expense of resulting in a solution that may not belong to our data points. To fix this, we introduce a second sub-problem that reimposes the underlying constraint and selects the datapoint that has the highest correlation with the point selected in the first sub-problem. These sub-problems are formulated as

$$(\boldsymbol{u}_k, \boldsymbol{v}_k) = \underset{\boldsymbol{u},\boldsymbol{v}}{\operatorname{argmin}} \ \|\boldsymbol{A} - \boldsymbol{U}_{\overline{k}}\boldsymbol{V}_{\overline{k}}^T - \boldsymbol{u}\boldsymbol{v}^T\|_F^2; \ \text{s.t. } \|\boldsymbol{u}\|_2 = 1, \tag{4a}$$

$$\mathbb{S}_k = \underset{m}{\operatorname{argmax}} \ |\boldsymbol{u}_k^T \tilde{\boldsymbol{e}}_m|. \tag{4b}$$

Here $\mathbb{S}_k$ is a singleton that contains the index of the selected data point. Matrices $\boldsymbol{U}_{\overline{k}}$ and $\boldsymbol{V}_{\overline{k}}$ are obtained by removing the k^{th} column of $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively. Sub-problem (4a) is equivalent to finding the first left singular vector (LSV) of $\boldsymbol{E}_{\overline{k}} \triangleq \boldsymbol{A} - \boldsymbol{U}_{\overline{k}}\boldsymbol{V}_{\overline{k}}^T$ and $\tilde{\boldsymbol{e}}_m$ is the normalized replica of the m^{th} column of the residual matrix, $\boldsymbol{E}_{\overline{k}}$. The set of normalized residuals is indicated by $\mathbb{E}_{\overline{k}}$. The constraint $\|\boldsymbol{u}\| = 1$ keeps $\boldsymbol{u}$ on the unit sphere to remove scale ambiguity between $\boldsymbol{u}$ and $\boldsymbol{v}$. Moreover, the unit sphere is a superset for $\mathbb{A}$ and keeps the modified problem close to

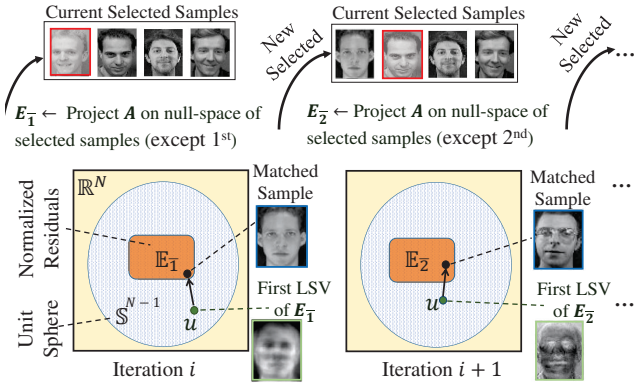

Figure 3: Two consecutive iterations of the SP algorithm. In each iteration the residual matrix, $\boldsymbol{E}_{\overline{k}}$, is computed. The first LSV of the residual matrix, $\boldsymbol{u}$, is a vector on $\mathbb{S}^{N-1}$. The most aligned column of $\boldsymbol{E}_{\overline{k}}$ with $\boldsymbol{u}$ is selected at each iteration and it is replaced for the next iteration. Please note that $\mathbb{E}_{\overline{k}} \subset \mathbb{S}^{N-1} \subset \mathbb{R}^N$ in the Venn diagrams.

the recast problem (3). After solving for $\boldsymbol{u}_k$, we find the data point that matches it the most in (4b). The steps of the SP algorithm are elaborated in Algorithm 1. Fig. 3 illustrates Problem (4) pictorially. SP is a low-complexity algorithm with no parameters to be tuned. The complexity order of computing the first singular component of an $M \times N$ matrix is $O(MN)$ [31]. As the proposed algorithm only needs the first singular component for each selection, the computational complexity of SP is $O(NM)$ per iteration which is much faster than convex relaxation-based algorithms with complexity $O(M^3)$ [19]. Moreover, SP performs faster than K-medoids algorithm and volume sampling, whose complexity is of order $O(KN(M-K)^2)$ and $O(MKN\log N)$, respectively [32, 33]. The stopping criterion can be convergence of set $\mathbb{S}$ or reaching a pre-defined maximum number of iterations. The convergence behavior of SP is studied in the supplementary document.

Simplicity and accuracy of SP facilitate its extension to nonlinear manifold sampling with a wide range of applications. We will refer to this extended version as kernel-SP (KSP) which is discussed next in Section 3.

3. Kernel SP: Selection based on a Locally Linear Model

The goal of CSSP introduced in (1) is to select a subset of data whose *linear subspace* spans all data. Obviously, this model is not proper for general data types that mostly lie on nonlinear manifolds. Accordingly, we generalize (1) and propose the following selection problem in order to efficiently sample from a union of manifolds

$$\underset{|\mathbb{S}|\leq K}{\operatorname{argmin}} \sum_{m=1}^{M} \|\boldsymbol{a}_m - \pi_{\mathbb{S}_m}(\boldsymbol{a}_m)\|_F^2 \;\; \text{s.t. } \mathbb{S}_m \subseteq \mathbb{S} \cap \Omega_m, \quad (5)$$

where Ω_m represent the indices of local neighbors of $\boldsymbol{a}_m$ based on an assumed distance metric. This problem is simplified to CSSP in Problem (1) if Ω_m is assumed to be equal

Algorithm 1 Spectrum Pursuit Algorithm

Require: $\boldsymbol{A}$ and K
Output: $\boldsymbol{A}_{\mathbb{S}}$
Initialization:
$\mathbb{S} \leftarrow$A random subset of $\{1,\dots,M\}$ with $|\mathbb{S}| = K$
$\{\mathbb{S}_k\}_{k=1}^{K} \leftarrow$Partition $\mathbb{S}$ into K subsets, each containing one element.
iter$=0$
while a stopping criterion is not met
  $k =$mod(iter,K)+1
  $\boldsymbol{U}_{\overline{k}} =$ normalize column$(\boldsymbol{A}_{\mathbb{S}\setminus\mathbb{S}_k})$
  $\boldsymbol{V}_{\overline{k}} = \boldsymbol{A}^T\boldsymbol{U}_{\overline{k}}(\boldsymbol{U}_{\overline{k}}^T\boldsymbol{U}_{\overline{k}})^{-1}$
  $\boldsymbol{E}_{\overline{k}} = \boldsymbol{A} - \boldsymbol{U}_{\overline{k}}\boldsymbol{V}_{\overline{k}}^T$ (null-space projection)
  $\boldsymbol{u}_k =$ find the first left singular vector of $\boldsymbol{E}_{\overline{k}}$ by solving (4a)
  $\mathbb{S}_k \leftarrow$ index of the most correlated column of $\boldsymbol{E}_{\overline{k}}$ with $\boldsymbol{u}_k$(4b)
  $\mathbb{S} \leftarrow \bigcup_{k'=1}^{K} \mathbb{S}_{k'}$
  iter=iter+1
end while

to $\{1,\cdots,M\}$. Problem (5) is written for each column of $\boldsymbol{A}$ separately in order to engage neighborhood for each data. This problem facilitates fitting a locally linear subspace for each data sample in terms of its neighbors. *Nonlinear* techniques demonstrates significant improvement upon linear methods for many scenarios [34, 35, 36].

Similar to Section 2, where we introduced SP as a low-complexity algorithm to tackle the NP-hard Problem (1), here we propose an extention of SP, referred to as kernel SP (KSP), to tackle the combinatorial search Problem (5). Manifold-based dimension reduction techniques and clustering algorithms do not provide prototypes suitable for data selection. However, inspired by spectral clustering of manifolds [37], main tool for nonlinear data analysis that partitions data into nonlinear clusters based on spectral components of the corresponding normalized similarity matrix, we formulate KSP as

$$\mathbb{S} = \underset{|\mathbb{S}|\leq K}{\operatorname{argmin}} \|\boldsymbol{L} - \pi_{\mathbb{S}}(\boldsymbol{L})\|_F^2, \quad (6)$$

where $\boldsymbol{L} = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{S}\boldsymbol{D}^{-\frac{1}{2}}$, is the normalized similarity matrix of the data. Matrix $\boldsymbol{S} = [s_{ij}] \in \mathbb{R}^{M\times M}$ is defined as the similarity matrix of data and $\boldsymbol{D}$ is a diagonal matrix and $d_{ii} = \sum_{j\neq i} s_{ij}$. The similarity matrix can be defined based on any similarity measure. A typical choice is a Guassian kernel with parameter α. Note that problem (6) is the same as problem (1), where $\boldsymbol{A}$ is replaced by $\boldsymbol{L}$. The steps of the KSP algorithm are summarized in Algorithm 2.

Algorithm 2 Kernel Spectrum Pursuit

Require: $\boldsymbol{A}$, α, and K
Output: $\mathbb{S}$
$\boldsymbol{S} \leftarrow$Similarity Matrix: $s_{ij} = e^{-\alpha\|\boldsymbol{a}_i - \boldsymbol{a}_j\|_2^2}$
Form diagonal matrix $\boldsymbol{D}$ where $d_{ii} = \sum_{i\neq j} \boldsymbol{s}_{ij}$
$\boldsymbol{L} = \boldsymbol{D}^{-1/2}\boldsymbol{S}\boldsymbol{D}^{-1/2}$.
$\mathbb{S} \leftarrow$ Apply SP on $\boldsymbol{L}$ with K (Alg. 1)

4. Empirical Results and Some Applications of SP/KSP

To evaluate the performance of our proposed selection algorithms we consider several applications and conduct extensive experiments. The selected applications in this paper are (i) fast GAN training using reduced dataset; (ii) semi-supervised learning on graph-based datasets; (iii) large graph summarization; (iv) few-shot learning; and (v) open-set identification.

4.1. Training GAN

There have been many efforts [38, 39] to employ manifold properties to stabilize GAN training process and improve the quality of generated samples but none of them benefit from smart samples selection to expedite the training as suggested in the first column of Fig. 1. Here, we present our experimental results on CMU Multi-PIE Face Database [40] for representative selection. We use 249 subjects with various poses, illuminations, and expressions. There are 520 images per subject. Fig. 4 (top) depicts 10 selected images of a subject based on different selection methods: SP (our proposed) is compared with three well-known selection algorithms, DS3 [41], VS [33], and K-medoids [42]. As it can be seen, SP selects from more diverse angles. Fig. 4 (bottom) compares the performance of different state-of-the-art selection algorithms in terms of normalized projection error of CSSP, which is defined as the cost function in (1). As shown, SP outperforms all other methods. There is also a considerable performance gap between SP and IPM [17], the second best algorithm.

Next, to investigate the impact of selection in a real application, we use the selected samples to train a generative adversarial network (GAN) to generate multi-view images from a single-view input. For that, the GAN architecture in [45] is employed. The experimental setup and the implementation details from [45] are used, where the first 200 subjects are used for training and the rest for testing.

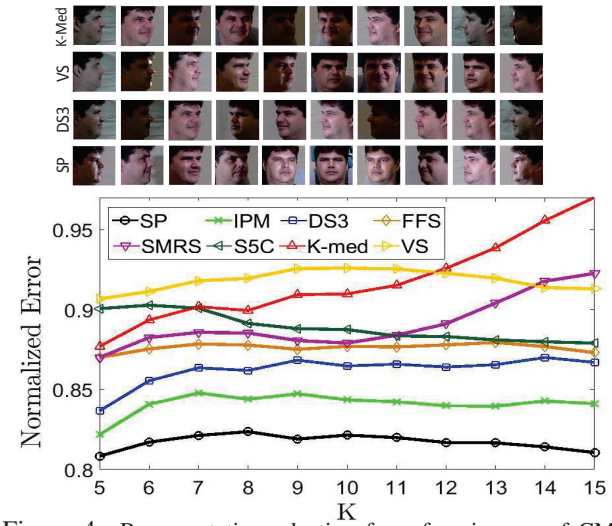

Figure 4: Representative selection from face images of CMU Multi-pie dataset. (Top) 10 images selected from 520 images of a subject. (Bottom) Averaged projection error for different number of representatives from 249 subjects. The projection error is normalized by projection error of random selection. The proposed SP algorithm is compared with IPM [17], DS3 [41], FFS [43], SMRS [27], S5C [44], K-medoids [42] and VS [33].

We select only 9 images from each subject and train the network with the selected images for 300 epochs using the batch size of 36. Table 1 shows the normalized ℓ_2 distances between features of the real and generated images, indicated as identity dissimilarities, averaged over all the images in the testing set. Features are extracted using a ResNet18 trained on MS-Celeb-1M dataset [46, 47]. As can be seen, SP and KSP outperform other selection methods. Moreover, KSP performs better than SP due to the selection from a nonlinear manifold.

The test set contains multi-view images from 50 subjects not seen in training. In the test phase, a single view from each of these 50 people is given and we are to generate other views. Please note that in this application, we *do have* ground truth images for all views. Hence, any similarity measure can be applied. The evaluation is performed identical to that of [45].

Table 1: Identity dissimilarities between real and GAN-generated images for different selection methods. For each method, GAN is trained based on the selected data points.

SMRS	S5C	FFS	DS3	K-Med	VS	IPM	SP	KSP
0.631	0.617	0.608	0.602	0.599	0.583	0.553	0.550	**0.546**
Trained GAN using All Data 0.5364								

4.2. Graph-based Semi-supervised Learning

To evaluate the performance of our proposed selection algorithm on more complicated scenarios, we consider the graph convolutional neural network (GCN) proposed in [48], that serves as a semi-supervised classifier on graph-based datasets. Indeed, a GCN takes a feature matrix and an adjacency matrix as inputs, and for every vertex of the graph produces a vector, whose elements correspond to the score of belonging to different classes. The semi-supervised task here considers the case where only a *selected* subset of nodes are labeled in the training set and the loss is computed based on the output vectors of these labeled nodes to perform back-propagation. Moreover, we inherit the same two-layer network architecture from [48]. To be more specific, an identity matrix is added to the original adjacency matrix so that every node is assigned with a self-connection. Further, we normalize the summation of two matrices using the kernel discussed in lines 2 and 3 of Algorithm 2 while the adjacency matrix serves as the similarity matrix S.

Our proposed KSP algorithm, together with other baselines, is tested on Cora dataset which is a real citation network dataset with $2,708$ nodes and $5,429$ number of edges as well as a random cluster-based graph datasets with 200

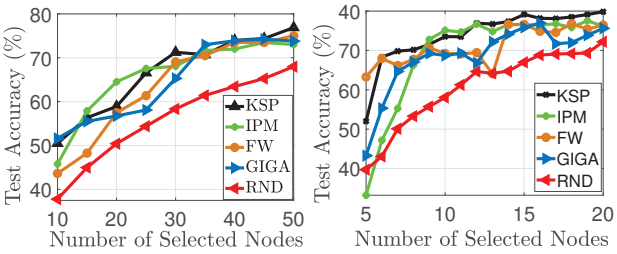

Figure 5: Semi-supervised classification accuracy of GCN on (Left) the Cora dataset [50]; and (Right) a random cluster-based graph dataset. Only the selected nodes are labeled and the subset selection is performed using the proposed KSP algorithm, and GIGA [51], FW [51], and random selection (RND).

nodes from 10 clusters, where every node independently belongs to one of the 10 clusters with equal probabilities and the cluster of a node serves as its label during the classification. Instead of constructing a completely connected graph, the existence of edges between node pairs is determined according to a density matrix, in which the density of edges connecting nodes in the same cluster is uniformly generated from the interval [0.2, 0.6], whereas the inter-cluster densities are randomly sampled according to a uniform distribution on the interval [0, 0.2]. The neural network is trained based on semi-supervised learning, i.e., the network is fed with the feature and adjacency matrices of the entire graph while the loss is only computed on the labeled vertices. Here the labeled vertices correspond to the subset of vertices that is selected by applying our proposed algorithm (KSP) on the normalized adjacency matrix. We train both datasets for a maximum of 100 epochs using Adam [49] with a learning rate of 0.01 and early stopping with a window size of 10, i.e. we stop training if the validation loss does not decrease for 10 consecutive epochs. The results are summarized in Figure 5. Due to the inherent randomness of training neural networks using gradient descent based optimizers, some ripples appear in the curves. However, it is clear hat, as expected, the test accuracy tends to increase as more labeled points are utilized for training. Further, as can be seen from the figure our proposed KSP algorithm significantly outperforms other algorithms for almost the whole range of selected points. This implies the superior performance of KSP in selecting the subset of data that comprises the most representative points of clusters. Lastly, because of the existence of outliers in a random graph, the accuracy of the proposed algorithm starts to improve slowly at about 70%, whereas other competitors saturate at about 60%. However, we note that the model is trained with only 10% of data, so this also implicitly suggests that our algorithm successfully picks out the most informative nodes.

4.3. Graph Summarization

Clusters (also known as communities) in a graph are those groups of vertices that share common properties. Identification of communities is a crucial task in graph-based systems. Instances include protein-protein interaction networks in biology [61], recommendation systems [62] in computer science, social media networks, etc. In the following, we design an experiment to find the vertices with a central position on several types of graphs, produced both by real datasets such as [55] and also synthetic graph which contains the aggregated network of some users' Facebook friends. In the later dataset, vertices represent individuals on Facebook, and edges between two users mean they are Facebook friends.

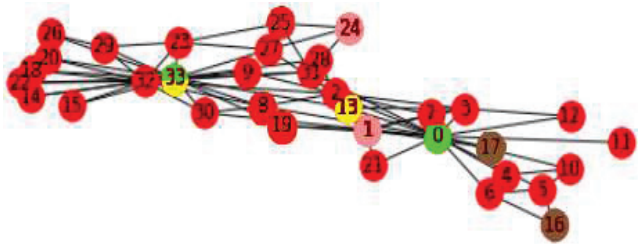

Figure 6: Zachary's Karate Club is a small social network where a conflict arises between the admin and the instructor in the club [60]. Each node of the club network represents a member of the karate club and a link between members indicate that they interact outside the club. The admin and the instructor which are the two nodes of this graph are $\{0, 33\}$, respectively. We apply KSP and two other algorithms to choose two of the main vertices. GIGA, MP and FW select ●, IS selects ●, VS selects ●, and KSP, FFS and DS3 select ●.

Various community detection based algorithms such as betweenness centrality (BC) has been proposed to measure the importance of a user in the network [54], by considering how many shortest paths pass through that user (vertex) for connecting each pair of other users (vertices). The more shortest paths that pass through the user, the more central the user is in the Facebook social network. Now assuming that a graph $\mathcal{G}$ or a similarity matrix is given, the aim is to first implement our method on the graph to approximate it with a subset of the vertices, and then exploit the measure of shortest path to evaluate the accuracy. We report the following performance measures: instead of computing the average shortest path between each vertex of the graph and all the other vertices which is really expensive (use of Dijkstra's algorithm n^2 times where n is the number of vertices), we compute the average shortest path between all the vertices and the selected vertices by KSP. The latter can be computed by using Dijkstra's algorithm only kn times, where k is the number of selected vertices.

Further, in this experiment we evaluate the performance of KSP compared to several state-of-the-art algorithms for data selection and coreset construction which is a small (weighted) subset of the data that approximates the full dataset. The results of these experiments are shown in Table 2 where 10 vertices from each graph are selected (except for Karate Club sketched in Fig. 6 from which we select 2 vertices) by different data selection algorithms. As can be seen our proposed method provides significant improvements in shortest path error over the state-of-the-art.

Table 2: Error performance of different state-of-the-art coreset construction algorithms for Graph summarization (central vertex selection) on various types of graphs. Practically all major social networks provide social clusters for instance, 'circles' on Google+, and 'lists' on Facebook and Twitter. For example, concerning Facebook ego graph, with KSP algorithm we define the task of identifying users' social clusters on a user's ego-network by exploiting the network structure.

Graph/Algorithm	RND	IS [52]	VS [33]	FFS [43]	MP [53]	DS3 [28]	IPM [17]	FW [51]	BC [54]	GIGA [51]	KSP
Facebook Ego [55]	0.2960	0.1250	0.2210	0.0142	0.0250	0.0147	0.0140	0.0190	0.0149	0.0145	**0.0130**
Powerlaw Cluster [56]	0.2739	0.2735	0.2732	**0.0167**	0.2701	0.0275	**0.0167**	0.2730	0.0358	0.0296	**0.0167**
Barabasi [57]	0.1630	0.1625	0.0142	0.0184	0.1625	0.0154	0.0156	0.1628	0.0378	0.0169	**0.0122**
Geo [58]	0.0685	0.0674	0.0683	0.0424	0.0493	0.0411	0.0299	0.0673	0.0014	0.0017	**0.0012**
Florentine [59]	0.0026	0.0006	0.0007	**0.0003**	0.001	0.0019	**0.0003**	0.0009	**0.0003**	**0.0003**	0.0004
Karate Club [60]	0.1388	0.0158	0.0326	**0.0117**	0.0146	**0.0117**	**0.0117**	0.0146	**0.0117**	0.0146	**0.0117**
Synthesized Graph	0.1421	0.1430	0.0115	0.0120	0.0143	0.0127	0.0122	0.0143	0.0143	0.0124	**0.0106**

4.4. Few Shot Learning

Training on Sampled Pairs: Next, we further evaluate the performance of SP on a more common data such as images and features. This analysis is motivated by the work in [63], as we employ their proposed neural network architecture named Siamese neural network. Moreover, we adopt the Omniglot dataset and split it into three subsets for training, validation, and test, each of which consists of totally different classes. For training and validation process two images are randomly sampled from their own corresponding data and are fed as the input to the Siamese neural network and a binary label is assigned to each pair according to the classes that they are sampled from. The network trained on these pairs achieves 90%+ accuracy in distinguishing inter-class and intra-class pairs.

Classification with Few-Shot Learning: After being fully trained on the sampled pairs, the model is developed for few-shot classification. In other words, if the model is accurate enough to distinguish the identity of classes to which the pairs belong to, given few representatives of a specific class, a trained Siamese network could serve as a binary classifier that verifies if the test instance belongs to this class. Therefore, the problem reduces to selecting the best representatives of every class to be paired with any test image. The class that produces the pairings with the highest average score is then identified as the classification result. The test set of Omniglot, after the partitioning discussed above, comprises 352 different classes, each of which is composed of 20 images. We sequentially evaluate every one of the 352 classes to choose the most informative subset of the 20 images by deploying our selection algorithm on the flattened features extracted from the last convolutional layer of the network that is fed with the 20 images. The classifier made from the Siamese network and the selected 352 representative groups are then evaluated on all the 7,000+ images in the test set. We show in Figure 7 the few-shot learning results when 2, 3, 4 and 5 images are selected out of the 20 images together with an example of selected groups in 2-shot learning.

It can be observed in Figure 7 that images selected by the evaluated algorithms are generally more standard and more

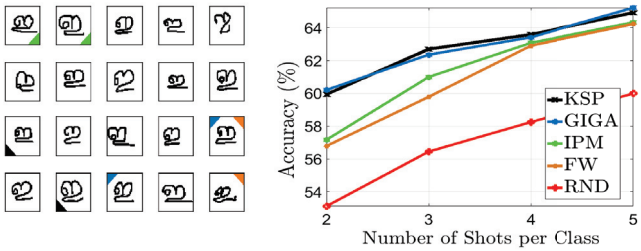

Figure 7: Learning of Omniglot's dataset on Siamese Neural Network using few shots. (Left) Visualization of selected images from the first class of Omniglot's test set in 2-shot learning. Images selected by an algorithm are marked in corners with the same color used in the right plot. (Right) Classification accuracy with few-shot learning.

identifiable than the others. Among all these competing algorithms, KSP makes the best selection for this character. Due to the fact that the classification accuracy is evaluated based on the 352 test classes, which do not appear in the training set, around 60% of correct classification is considerably acceptable. In particular, SP achieves accuracies of 59.84%, 62.70%, 63.55%, and 64.89% for 2-shot, 3-shot, 4-shot, and 5-shot classifications, respectively, which is comparable to the GIGA results of 60.21%, 62.36%, 63.42%, and 65.21% while outperforming other baseline algorithms. Note that SP needs less memory requirement and its computational complexity is less than its peers.

4.5. Open-Set Identification

In this experiment, the open-set identification problem is addressed employing propose selection method, which results in significant accuracy improvement compared to the state-of-the-art. In open-set identification, test data of a classification problem may come from unknown classes other than the classes employed during training, and the goal is to identify such samples belong to open-set and not the known labeled classes [64]. Interested readers are referred to [65, 66, 67, 68, 69] to the state-of-the-art approaches for solving open-set problem.

Employing the entire closed-set data during the training procedure leads to inclusion of untrustworthy samples of the closed-set. Regularized or underfitting models (such as low-rank representations [70, 71, 72]) still suffer from memorizing effect of such samples, which exacerbate the

separation between open and closed-set by adding ambiguity to the decision boundary between the closed and open-set classes. To resolve this issue, we utilize our proposed selection method, KSP, which selects the core representatives. Therefore, selected representatives for open-set identification are more robust in rejecting open-set test samples which do not fit well to the core representatives. We pictorially illustrate the proposed scheme for open-set identification in Fig. 1 on the rightmost panel and the proposed algorithm which is referred to as selection-based open-set identification scheme (SOSIS) hereunder (Algorithm 3).

Experiment Set-up: We use MNIST dataset as the closed-set with samples from Omniglot as the open-set. The ratio of Omniglot to MNIST test dataset is set to $1:1$ (10,000 from each), same as the simulation scenario in [67]. A classifier with ResNet-164 architecture [73] is trained on MNIST as for step 1 in Alg. 3. Results of macro-averaged F1-score [74] for SOSIS method with different selection methods and different number of samples are listed in Table 3 as well as the sate-of-the-art in [67]. The best achieved F1-score is 0.964 belonging to SOSIS with KSP selection using 50 representatives. The second best performance is by SOSIS with SP selection again using 50 representatives. Performance downgrade is observed for both scenarios of choosing too few representatives such as 5 or fewer and obsessively choosing all data.

The gap between the error values resulting from projection of open and closed-set onto selected samples computed in step 4 of Alg. 3 differs significantly compared to that of the projection onto the entire dataset (due to overfitting and memorization effect). We call this splitting property as reflected in Fig. 8 (a) (entire dataset) vs. 8 (b) (selected samples) at the testing phase. For a better visualization, projection errors are sorted separately for closed-set and open-set data at the testing phase. As observed, fewer number of representatives results in higher projection error. However, at the same time *closed-set and open-set test data are better split* as also observed in Fig. 8.

5. Conclusion

A novel approach to data selection from linear subspaces is proposed and its extension for selection from nonlinear manifolds is presented. The proposed SP algorithm demonstrates an accurate solution for CSSP. Moreover, SP and KSP have shown superior performance in many applications. The investigated fast and efficient deep learning frameworks, empowered by our selection methods, have shown that dealing with selected representatives is not only fast but can also be more effective. This manuscript is allocated mostly for algorithm designs and applications of data selection. Theoretical results and more buttressing experiments can be found in the supplementary document.[1]

Algorithm 3 Selection-based open-set identification (SOSIS)

Require: $\boldsymbol{A}^X$ (closed-set training data), and $\boldsymbol{A}^Y=\{\boldsymbol{a}_p^Y\}_{p=1}^P$ (test set)

1: Train a classifier on $\boldsymbol{A}^X$ on H classes
2: $\mathbb{S}_h \leftarrow$ set of K selected samples for class $\#h$ in $\boldsymbol{A}^X$
3: $\ell(p) \leftarrow$ label $\boldsymbol{a}_p^Y$ using trained classifier in Step 1 $(\forall p)$
4: $\text{err}(p) = \|\boldsymbol{a}_p^Y - \pi_{\mathbb{S}_{\ell(p)}}(\boldsymbol{a}_p^Y)\|_2$ $(\forall p)$
5: partition **err** to 2 sets using kmeans to set the threshold value thr

Output: open-set$\leftarrow \{\boldsymbol{a}_p^Y \mid \textbf{err}(p) \geq thr\}$

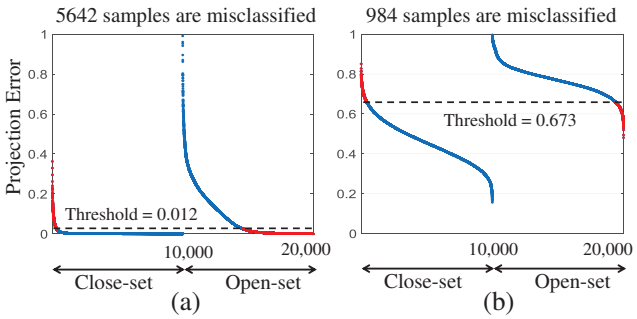

Figure 8: Sorted values of **err** in Step 4 of Alg. 3 for 20,000 test samples (10,000 per each closed/open set). (a) all data are selected as representatives. (b) only 20 representatives are selected. For both (a) and (b), a projection error above/below the threshold leads to classifying a sample as open-set/closed-set. Blue and red points correspond to the correctly-classified and missclassified samples, respectively. As shown, implementing SOSIS enabled by KSP has significantly reduced the number of misclassified samples, from 5642 to 984.

Table 3: Comparing F1-score of the proposed SOSIS algorithm with state-of-the-art methods for open-set identification. SP, KSP and FFS [43] are employed as the core of SOSIS.

method/K	5	20	50	100	500	All Data
SOSIS (based on FFS)	0.876	0.913	0.944	0.952	0.841	0.792
SOSIS (based on SP)	0.904	0.945	0.958	0.952	0.824	
SOSIS (based on KSP)	0.928	0.959	**0.964**	0.959	0.827	

Supervised only [67]	0.680
LadderNet [67]	0.764
DHRNet [67]	0.793

6. Acknowledgements

This research is based upon work supported in parts by the National Science Foundation under Grants No. 1741431 and CCF-1718195 and the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. D17PC00345. Authors appreciate valuable comments from Alireza Zaeemzadeh, Marziyeh Edraki and Mehrdad Salimitari. The views, findings, opinions, and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the NSF, ODNI, IARPA, or the U.S. Government.

[1]The authors release their codes in the following anonymous link "https://github.com/cvpr4942/CVPR-2020-Conf.git".

References

[1] UK. ATT Laboratories, Cambridge. The orl database of faces (now att the database of faces. In *Available online: https://git-disl.github.io/GTDLBench/datasets/att$_face_dataset$/*, 1994.

[2] Ali Çivril. Column subset selection problem is ug-hard. *Journal of Computer and System Sciences*, 80(4):849–859, 2014.

[3] Christos Boutsidis, Michael W Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 968–977. SIAM, 2009.

[4] Yaroslav Shitov. Column subset selection is np-complete. *arXiv preprint arXiv:1701.02764*, 2017.

[5] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Math. Program.*, 14(1):265–294, 1978.

[6] Tony F Chan and Per Christian Hansen. Some applications of the rank revealing qr factorization. *SIAM Journal on Scientific and Statistical Computing*, 13(3):727–741, 1992.

[7] Tony F Chan. Rank revealing qr factorizations. *Linear algebra and its applications*, 88:67–82, 1987.

[8] Jianwei Xiao, Ming Gu, and Julien Langou. Fast parallel randomized qr with column pivoting algorithms for reliable low-rank matrix approximations. In *2017 IEEE 24th International Conference on High Performance Computing (HiPC)*, pages 233–242. IEEE, 2017.

[9] Venkatesan Guruswami and Ali Kemal Sinop. Optimal column-based low-rank matrix reconstruction. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1207–1214. SIAM, 2012.

[10] Chengtao Li, Stefanie Jegelka, and Suvrit Sra. Polynomial time algorithms for dual volume sampling. In *Advances in Neural Information Processing Systems*, pages 5038–5047, 2017.

[11] Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1117–1126. Society for Industrial and Applied Mathematics, 2006.

[12] Saurabh Paul, Malik Magdon-Ismail, and Petros Drineas. Column selection via adaptive sampling. In *Advances in neural information processing systems*, pages 406–414, 2015.

[13] Yining Wang and Aarti Singh. Provably correct algorithms for matrix column subset selection with selectively sampled data. *Journal of Machine Learning Research*, 18:1–42, 2018.

[14] Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near-optimal column-based matrix reconstruction. *SIAM Journal on Computing*, 43(2):687–717, 2014.

[15] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[16] Michal Derezinski, Manfred K Warmuth, and Daniel J Hsu. Leveraged volume sampling for linear regression. In *Advances in Neural Information Processing Systems*, pages 2505–2514, 2018.

[17] Alireza Zaeemzadeh, Mohsen Joneidi, Nazanin Rahnavard, and Mubarak Shah. Iterative Projection and Matching: Finding Structure-Preserving Representatives and Its Application to Computer Vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5414–5423, 2019.

[18] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[19] Siddharth Joshi and Stephen Boyd. Sensor Selection via Convex Optimization. *IEEE Transactions on Signal Processing*, 57(2):451–462, 2 2009.

[20] Ali Çivril and Malik Magdon-Ismail. On selecting a maximum volume sub-matrix of a matrix and related problems. *Theoretical Computer Science*, 410(47-49):4801–4811, 2009.

[21] Zelda E Mariet and Suvrit Sra. Elementary symmetric polynomials for optimal experimental design. In *Advances in Neural Information Processing Systems*, pages 2139–2148, 2017.

[22] Haim Avron and Christos Boutsidis. Faster subset selection for matrices and applications. *SIAM Journal on Matrix Analysis and Applications*, 34(4):1464–1499, 2013.

[23] Mohsen Joneidi, Alireza Zaeemzadeh, Behzad Shahrasbi, Guo-Jun Qi, and Nazanin Rahnavard. E-optimal Sensor Selection for Compressive Sensing-based Purposes. *IEEE Transactions on Big Data*, page To Appear in, 2018.

[24] Aleksandar Nikolov, Mohit Singh, and Uthaipon Tao Tantipongpipat. Proportional volume sampling and approximation algorithms for a-optimal design. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1369–1386. SIAM, 2019.

[25] Michal Derezinski and Manfred Warmuth. Subsampling for Ridge Regression via Regularized Volume Sampling. In *International Conference on Artificial Intelligence and Statistics*, pages 716–725, 2018.

[26] Chen Dan, Hong Wang, Hongyang Zhang, Yuchen Zhou, and Pradeep K Ravikumar. Optimal analysis of subset-selection based l_p low-rank approximation. In *Advances in Neural Information Processing Systems*, pages 2537–2548, 2019.

[27] Ehsan Elhamifar, Guillermo Sapiro, and Rene Vidal. See all by looking at a few: Sparse modeling for finding representative objects. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1600–1607. IEEE, 2012.

[28] Ehsan Elhamifar, Guillermo Sapiro, and S. Shankar Sastry. Dissimilarity based sparse subset selection. *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, 38(11):2182–2197, 2016.

[29] Jingjing Meng, Hongxing Wang, Junsong Yuan, and Yap-Peng Tan. From Keyframes to Key Objects: Video Summarization by Representative Object Proposal Selection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1039–1048. IEEE, 6 2016.

[30] Huaping Liu, Yunhui Liu, and Fuchun Sun. Robust Exemplar Extraction Using Structured Sparse Coding. *IEEE Transactions on Neural Networks and Learning Systems*, 26(8):1816–1821, 8 2015.

[31] P Comon and G H Golub. Tracking a few extreme singular values and vectors in signal processing. *Proceedings of the IEEE*, 78(8):1327–1343, 1990.

[32] P A Vijaya, M Narasimha Murty, and D K Subramanian. Leaders–Subleaders: An efficient hierarchical clustering algorithm for large data sets. *Pattern Recognition Letters*, 25(4):505–513, 2004.

[33] Amit Deshpande and Luis Rademacher. Efficient volume sampling for row/column subset selection. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 329–338. IEEE, 2010.

[34] Dhish Kumar Saxena and Kalyanmoy Deb. Non-linear dimensionality reduction procedures for certain large-dimensional multi-objective optimization problems: Employing correntropy and a novel maximum variance unfolding. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 772–787. Springer, 2007.

[35] Ming-Hsuan Yang. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. In *Fgr*, volume 2, page 215, 2002.

[36] Mahlagha Sedghi, George Atia, and Michael Georgiopoulos. A multi-criteria approach for fast and outlier-aware representative selection from manifolds. *arXiv preprint arXiv:2003.05989*, 2020.

[37] Yong Wang, Yuan Jiang, Yi Wu, and Zhi-Hua Zhou. Spectral clustering on multiple manifolds. *IEEE Transactions on Neural Networks*, 22(7):1149–1161, 2011.

[38] Marzieh Edraki and Guo-Jun Qi. Generalized loss-sensitive adversarial learning with manifold margins. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 87–102, 2018.

[39] Marzieh Edraki, Nazanin Rahnavard, and Mubarak Shah. Subspace capsule network. *arXiv preprint arXiv:2002.02924*, 2020.

[40] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-PIE. *Image and Vision Computing*, 28(5):807–813, 5 2010.

[41] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781, 2013.

[42] P A Vijaya, M Narasimha Murty, and D K Subramanian. Leaders–Subleaders: An efficient hierarchical clustering algorithm for large data sets. *Pattern Recognition Letters*, 25(4):505–513, 2004.

[43] Chong You, Chi Li, Daniel P Robinson, and René Vidal. Scalable exemplar-based subspace clustering on class-imbalanced data. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 67–83, 2018.

[44] Shin Matsushima and Maria Brbic. Selective sampling-based scalable sparse subspace clustering. In *Advances in Neural Information Processing Systems*, 2019.

[45] Yu Tian, Xi Peng, Long Zhao, Shaoting Zhang, and Dimitris N. Metaxas. CR-GAN: Learning Complete Representations for Multi-view Generation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 942–948, California, 7 2018. International Joint Conferences on Artificial Intelligence Organization.

[46] Kaidi Cao, Yu Rong, Cheng Li, Xiaoou Tang, and Chen Change Loy. Pose-Robust Face Recognition via Deep Residual Equivariant Mapping. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[47] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. MS-Celeb-1M: Challenge of Recognizing One Million Celebrities in the Real World. *Electronic Imaging*, 2016(11):1–6, 2 2016.

[48] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

[49] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[50] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. Technical report, 2008.

[51] Trevor Campbell and Tamara Broderick. Bayesian coreset construction via greedy iterative geodesic ascent. In *International Conference on Machine Learning*, 2018.

[52] Andreas Krause Olivier Bachem, Mario Lucic. Practical coreset constructions for machine learning. *Thesis at Department of Computer Science, ETH Zurich*, 2017.

[53] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Hierarchical matching pursuit for image classification: Architecture and fast algorithms. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 2115–2123, 2011.

[54] Santo Fortunato. Community detection in graphs. *CoRR*, abs/0906.0612, 2009.

[55] Jure Leskovec and Julian J. Mcauley. Learning to discover social circles in ego networks. In *Advances in Neural Information Processing Systems 25*, pages 539–547. 2012.

[56] William Aiello, Fan Chung Graham, and Linyuan Lu. A random graph model for power law graphs. *Experimental Mathematics*, 10(1):53–66, 2001.

[57] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[58] Aric Hagberg Milan Bradonjic and Allon George Percus. Giant component and connectivity in geographical threshold graphs. *in Algorithms and Models for the Web-Graph (WAW), Antony Bonato and Fan Chung (Eds)*, pages 209–216, 9 2007.

[59] Ronald L. Breiger and Philippa E. Pattison. Cumulated social roles: The duality of persons and their algebras. *Social Networks*, 8, 9 1986.

[60] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[61] Jingchun Chen and Bo Yuan. Detecting functional modules in the yeast protein-protein interaction network. *Bioinformatics*, 22(18):2283–2290, 2006.

[62] Wang-Cheng Kang, Eric Kim, Jure Leskovec, Charles Rosenberg, and Julian J. McAuley. Complete the look: Scene-based complementary product recommendation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 10532–10541, 2019.

[63] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. 2015.

[64] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016.

[65] Qianyu Feng, Guoliang Kang, Hehe Fan, and Yi Yang. Attract or distract: Exploit the margin of open set. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7990–7999, 2019.

[66] Haipeng Xiong, Hao Lu, Chengxin Liu, Liang Liu, Zhiguo Cao, and Chunhua Shen. From open set to closed set: Counting objects by spatial divide-and-conquer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8362–8371, 2019.

[67] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4016–4025, 2019.

[68] Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. Iterative learning with open-set noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8688–8696, 2018.

[69] Trung Pham, Vijay BG Kumar, Thanh-Toan Do, Gustavo Carneiro, and Ian Reid. Bayesian semantic instance segmentation in open set world. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–18, 2018.

[70] A. Esmaeili and F. Marvasti. A novel approach to quantized matrix completion using huber loss measure. *IEEE Signal Processing Letters*, 26(2):337–341, 2019.

[71] Ashkan Esmaeili, Kayhan Behdin, Mohammad Amin Fakharian, and Farokh Marvasti. Transductive multi-label learning from missing data using smoothed rank function. *Pattern Analysis and Applications*, pages 1–9, 2020.

[72] Mehrdad Salimitari, Mohsen Joneidi, and Mainak Chatterjee. Ai-enabled blockchain: An outlier-aware consensus protocol for blockchain-based iot networks. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2019.

[73] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[74] Zachary C Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. Optimal thresholding of classifiers to maximize f1 measure. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 225–239. Springer, 2014.