5-27-2022

Low-Rank Clustering via LP1-PCA

Matt Krol
mrk7339@rit.edu

Low-Rank Clustering via LP1-PCA

by

Matt Krol

Thesis
Submitted in Partial Fulfillment of the Requirements for the Degree of
MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

Department of Electrical and Microelectronic Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology, Rochester, NY

Supervised by
Dr. Panos Markopoulos

May 27, 2022

Approved by:

Dr. Panos Markopoulos
Associate Professor, Department of Electrical and Microelectronic Engineering
Primary Thesis Advisor

Dr. Sohail Dianat
Professor, Department of Electrical and Microelectronic Engineering
Committee Member

Dr. Andreas Savakis
Professor, Department of Computer Engineering
Committee Member

Dr. Ferat Sahin
Professor, Department of Electrical and Microelectronic Engineering
Department Head

Abstract

In recent years, subspace clustering has found many practical use cases which include, for example, image segmentation, motion segmentation, and facial clustering. The image and video data that is common to these types of applications often has high dimensionality. Rather than viewing high dimensionality as a drawback, we propose a novel algorithm for subspace clustering that takes advantage of the high dimensional nature of such data. We call this algorithm LP1-PCA Spectral Clustering. Specifically, we introduce a concept that we call cluster-ID sparsity, and we propose an algorithm called LP1-PCA to attain this in low data dimensions. Our novel LP1-PCA algorithm is simple to implement and typically converges after only a few iterations. Conditions for which our algorithm performs well are discussed both theoretically and empirically, and we show that our method often attains superior clustering performance when compared to other common clustering algorithms on synthetic and real world datasets.

Acknowledgment

I would like to thank my advisor Dr. Panos Markopoulos for his support and guidance throughout the course of this project. In addition, I would like to thank Dimitris Chachlakis for his contributions to this project while it was in its early stages. Finally, I would like to thank the Air Force Research Laboratory (AFRL) and the National Science Foundation (NSF) for the funding that made this work possible.

Contents

List of Figures

List of Tables

1 Introduction

Modern big data typically reside in very high ambient dimensions, e.g., images and videos. It is well known however that such data often have low-rank structure that can be utilized. This structure can be exploited by using methods such as Principal Component Analysis (PCA) and Fisher's Linear Discriminant (LDA) to pre-process data. The PCA problem attempts to find a projection onto a lower dimensional subspace that best represents the data in the squared error sense. Methods like LDA also try to project the data onto a lower dimensional subspace, but do so by maximizing the ratio of between class variance and inter-class variance to maximize class separation in the new low-rank space. Such techniques have made processing high dimensional data feasible for many applications in signal processing, statistics, pattern recognition, etc.

The low-rank structure hidden in high dimensional data can also be used in unsupervised clustering. For example, if given an image, one might want to segment the image into categories or clusters. Such problems are solved by subspace clustering techniques. The idea is that images, and other high dimensional data, are often well described by affine subspaces present in the data. Therefore, a subspace clustering algorithm aims to find the parameters of each subspace along with the subspace memberships of each data sample. As described earlier, one common method of processing high dimensional data in a machine learning model is to perform dimensionality reduction using PCA before solving the problem at hand. In this work, we will show that it is possible to utilize the high-dimensional nature of the data in subspace clustering problems.

There are many approaches to developing subspace clustering algorithms. One approach is to use matrix factorization methods as in the algorithms of Boult and Brown [1], Costeira and Kanade [2], and Gear [3]. Algebraic methods like Generalized Principal Component Analysis (GPCA) [4] also exist. Another approach to subspace clustering is to use statistical based methods. Some examples of such algorithms are PPCA [5], ALC [6], and RANSAC [7]. Our work focuses on forming subspace affinities that can used in spectral clustering. The

spectral clustering framework as described in the work of Ng et al. [8] has a foundational role in many existing subspace clustering algorithms, e.g., in Sparse Subspace Clustering (SSC) [9], Spectral Curvature Clustering (SCC) [10], Local Subspace Affinity (LSA) [11], Spectral Local Best-Fit Flats (SLBF), Locally Linear Manifold Clustering (LLMC) [12], and Low-Rank Representation (LRR) [13]. For a more extensive survey of subspace clustering algorithms, we direct the reader to the work of Vidal [14].

In our work, we are inspired by the observation that high dimensional low-rank data tends to promote orthogonality. We therefore introduce a property that we call cluster-ID sparsity and propose a novel algorithm called LP1-PCA to obtain this in low data dimension. In addition, we propose a novel subspace clustering algorithm called LP1-PCA Spectral Clustering that utilizes cluster-ID sparsity to form subspace affinities when the data has low-rank and high dimension. Our novel LP1-PCA Spectral Clustering algorithm is simple to implement, converges quickly, and often attains superior clustering performance when compared to other common clustering algorithms on synthetic and real world datasets.

In summary, the contributions that we make in this work are as follows:

- We show empirical evidence that uncorrelated low-rank data from different subspaces tend to be nearly orthogonal when the ambient dimension is high.
- We show empirical evidence that uncorrelated low-rank data from different subspaces can be used to form subspace affinities that will achieve perfect clustering via spectral clustering via spectral clustering.
- We formulate the LP1-PCA problem as a novel sparsity promoting PCA variant and we solve it exactly for rank-1 analysis and any $p \geq 1$.
- We propose an approximate iterative algorithm for LP1-PCA for general rank and any $p \geq 1$.
- We propose a method of forming sparse subspace affinities using LP1-PCA when the data is high dimensional and predominantly low-rank.

- We show empirical evidence over an array of synthetic and real-world data that the proposed sparse subspace affinities obtained by means of the proposed LP1-PCA method often outperform standard clustering algorithms.

2 Background

2.1 Notation

This section will clarify the notation used throughout the rest of this paper. The script letter $\mathbb{R}$ denotes the set of real numbers and the script letter $\mathbb{N}$ denotes the set of natural numbers excluding zero. Accordingly, the set of real valued $m \times n$ matrices is denoted by $\mathbb{R}^{m \times n}$ and the set of real valued $n \times 1$ column vectors is denoted by $\mathbb{R}^n$. Matrices and column vectors are notated using bold upper and lower case letters, respectively. To simplify notation, we will find it useful to use the hadamard product of two matrices and the hadamard power of a matrix. For any two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$ the hadamard product of $\mathbf{A}$ and $\mathbf{B}$ is defined as $\mathbf{A} \circ \mathbf{B} = \mathbf{C} \in \mathbb{R}^{m \times n}$ where $\mathbf{C}_{i,j} = \mathbf{A}_{i,j}\mathbf{B}_{i,j}$. For any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $x \in \mathbb{R}$ the hadamard power of a matrix $\mathbf{A}$ is defined as $\mathbf{A}^{\circ x} = \mathbf{Y} \in \mathbb{R}^{m \times n}$ where $\mathbf{Y}_{i,j} = \mathbf{A}_{i,j}^x$. In this work, we will frequently utilize matrices in the $d \times k$ Steifel manifold which is defined as $\mathcal{S}_{d \times k} = \left\{ \mathbf{Q} \in \mathbb{R}^{d \times k} : \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_k \right\}$.

2.2 Norms and Hölder's Inequality

A norm is a measure of the size of a vector or matrix containing elements in $\mathbb{R}$. Accordingly, norms also induce a measure of distance between two pairs of vectors or matrices. In this work, we will make heavy use of the family of L_p-norms, which are generalizations of the euclidean norm. For our purposes, we will utilize the L_p-norm family for their sparsity and density inducing effects. We say that a matrix or vector is sparse if most of its entries are zero. Conversely, we say that a matrix or vector is dense if most of its entries are non-zero.

Definition 2.1 (Vector L_p-norm)**.** *For any vector* $\mathbf{y} \in \mathbb{R}^n$ *and* $p \in \mathbb{R} \geq 1$*, the* L_p*-norm of* $\mathbf{y}$ *is defined as*

$$\|\mathbf{y}\|_p = \left(\sum_{i=1}^{m} |y_i|^p \right)^{1/p}. \tag{1}$$

Fig. 1 shows the boundaries of the L_p-norm unit ball, i.e., the set $\mathcal{B}_p^d = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_p \leq 1\}$ with $d = 2$, and different values of p.

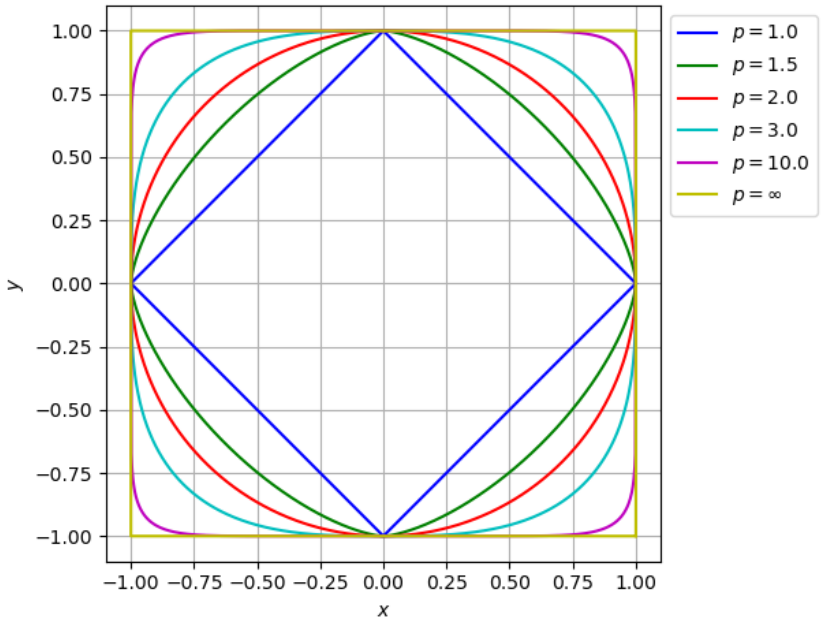

Figure 1: L_p-norm unit ball for different values of p.

From Fig. 1, the circular shape of the L_2-norm corresponds directly to that of the standard euclidean norm. In addition, we observe that $\lim_{p\to\infty} \|\mathbf{x}\|_p = \max_{1\leq i\leq d} |x_i|$, which results in a box shape. The L_1-norm results in a diamond shape.

When working on mathematical problems that involve vectors, it is common to encounter a need for the Cauchy-Schwartz Inequality. However, for general L_p-norms, the Cauchy-Schwartz Inequality is not usable. Fortunately, there is an alternative inequality that is valid for the entire family of L_p-norms.

Theorem 2.1 (Hölder's Inequality)**.** *Let* $\mathbf{v}, \mathbf{w} \in \mathbb{R}^n$ *and* $p, q \in \mathbb{R} \geq 1$*. If* $\frac{1}{p} + \frac{1}{q} = 1$*, then*

$$\left|\mathbf{v}^T \mathbf{w}\right| \leq \|\mathbf{w}\|_p \|\mathbf{v}\|_q . \tag{2}$$

Proof. A proof of Hölder's Inequality can be found in most functional analysis textbooks, e.g., the work in [15]. □

Without loss of generality, in Theorem 2.1, it suffices to fix $p \geq 1$ and set $q = p/(p-1)$. It is easy to verify that setting $p = 2$ implies that $q = 2$, which reduces Theorem 2.1 to the Cauchy-Schwartz Inequality. Furthermore, setting $p = 1$ implies that $q = \infty$, and so on. This relationship suggests that there is a dualistic nature between p and q, and for this reason, p and q are sometimes referred to as dual exponents. Before we explore the nature of this duality, we will discuss a few consequences of Theorem 2.1 that we will make use of in this work.

Corollary 2.1.1. *For any* $\mathbf{x} \in \mathbf{R}^n$ *and* $q \geq p \geq 1$ *we have* $\|\mathbf{x}\|_q \leq \|\mathbf{x}\|_p$.

Proof. This result is proven in most functional analysis textbooks, e.g., the work in [15]. □

Corollary 2.1.2. *Suppose that* $\mathbf{v}, \mathbf{w} \in \mathbb{R}^n$ *and* $p \in \mathbb{R} \geq 1$*, then the inequality in Theorem 2.1 can achieve equality when* $v_i = \operatorname{sgn}(w_i) |w_i|^{p-1}$ *for all* $i \in \{1, 2, \ldots, n\}$.

Proof. By Theorem 2.1, it holds that

$$\left|\mathbf{v}^T \mathbf{w}\right| \leq \|\mathbf{w}\|_p \|\mathbf{v}\|_{p/(p-1)} . \tag{3}$$

Next, we expand the norms, which results in

$$\left|\sum_{i=1}^{n} v_i w_i\right| \leq \left(\sum_{i=1}^{n} |w_i|^p\right)^{1/p} \left(\sum_{i=1}^{n} |v_i|^{p/(p-1)}\right)^{(p-1)/p} . \tag{4}$$

Replacing v_i with $\operatorname{sgn}(w_i)\,|w_i|^{p-1}$ yields,

$$\left|\sum_{i=1}^{n} \operatorname{sgn}(w_i)\,|w_i|^{(p-1)}\,w_i\right| \leq \left(\sum_{i=1}^{n} |w_i|^p\right)^{1/p} \left(\sum_{i=1}^{n} \left|\operatorname{sgn}(w_i)\,|w_i|^{(p-1)}\right|^{p/(p-1)}\right)^{(p-1)/p} \tag{5}$$

$$\left|\sum_{i=1}^{n} |w_i|\,|w_i|^{(p-1)}\right| \leq \left(\sum_{i=1}^{n} |w_i|^p\right)^{1/p} \left(\sum_{i=1}^{n} |w_i|^p\right)^{(p-1)/p} \tag{6}$$

$$\left|\sum_{i=1}^{n} |w_i|^p\right| \leq \sum_{i=1}^{n} |w_i|^p\,. \tag{7}$$

The outermost absolute value on the left hand side is redundant, so we have

$$\sum_{i=1}^{n} |w_i|^p = \sum_{i=1}^{n} |w_i|^p\,, \tag{8}$$

which gives us the desired result. □

Corollary 2.1.2 shows that the upper bound of the inequality in Theorem 2.1 is achievable. As a consequence, we can derive another useful result.

Corollary 2.1.3. *For any $p \in \mathbb{R} \geq 1$ and $\mathbf{w} \in \mathbb{R}^n$, the L_p-norm of $\mathbf{w}$ can be expressed as*

$$\|\mathbf{w}\|_p = \max_{\mathbf{v}} \frac{\mathbf{v}^T\mathbf{w}}{\|\mathbf{v}\|_q}, \tag{9}$$

where $q = p/(p-1)$. Furthermore, an optimal $\mathbf{v}$ is given by Corollary 2.1.2.

Proof. This is a direct result of Theorem 2.1 and Corollary 2.1.2. □

Corollary 2.1.3 gives us just one example of the dualistic nature between p and q. Another example of the dualistic nature between p and q can be shown by considering the following pair of optimization problems

$$\arg\max_{\|\mathbf{x}\|_2=1} \|\mathbf{x}\|_p \tag{10}$$

and

$$\arg\min_{\|\mathbf{x}\|_2=1} \|\mathbf{x}\|_p. \tag{11}$$

Fig. 2 shows the image of the feasibility set under the objective function for both (10) and (11) in two dimensions, which is expressed as $\|\mathbf{x}(\theta)\|_p$ with $\mathbf{x}(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \end{bmatrix}^T$.

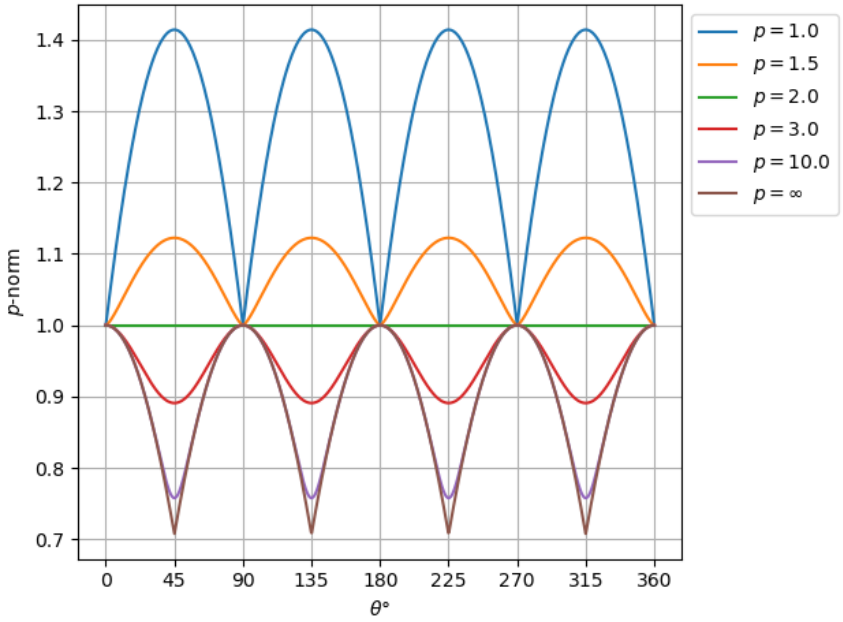

Figure 2: Image of the two-dimensional L_2 unit circle under different L_p-norms.

From Fig. 2, we can make the following observations:

i. When $1 \leq p < 2$, the maximization problem in (10) is solved by $\begin{bmatrix} \pm\frac{\sqrt{2}}{2} & \pm\frac{\sqrt{2}}{2} \end{bmatrix}^T$.

ii. When $1 \leq p < 2$, the minimization problem in (11) is solved by $\begin{bmatrix} \pm 1 & 0 \end{bmatrix}^T$ or $\begin{bmatrix} 0 & \pm 1 \end{bmatrix}^T$.

iii. When $2 < p < \infty$, the maximization problem in (10) is solved by $\begin{bmatrix} \pm 1 & 0 \end{bmatrix}^T$ or $\begin{bmatrix} 0 & \pm 1 \end{bmatrix}^T$.

iv. When $2 < p < \infty$, the minimization problem in (11) is solved by $\begin{bmatrix} \pm\frac{\sqrt{2}}{2} & \pm\frac{\sqrt{2}}{2} \end{bmatrix}^T$.

These observations suggest that for $1 \leq p < 2$, the solution to (10) is dense and the solution to (11) is sparse. On the contrary, for $p > 2$, the solution to (10) is sparse and the solution to (11) is dense. Furthermore, Fig. 2 suggests that

$$\arg\max_{\|\mathbf{x}\|_2=1} \|\mathbf{x}\|_p = \arg\min_{\|\mathbf{x}\|_2=1} \|\mathbf{x}\|_q, \tag{12}$$

for any $p \geq 1$ and $q = p/(p-1)$. This observation is notable since most practitioners in the fields of signal processing, machine learning, and statistics are accustomed to associating sparsity with the L_1-norm. This is likely because the problems in these fields are most often structured as minimization problems, e.g., LASSO Regression [16], Basis Pursuit [17], Compressed Sensing [18], etc. In this work, we consider maximization problems involving various L_p-norms, which implies that the L_1-norm will induce density.

Lastly, we introduce an entry-wise matrix norm that will be utilized in this work.

Definition 2.2 (Matrix $L_{p,q}$-norm)**.** *For any matrix $X \in \mathbb{R}^{m\times n}$ and $p, q \in \mathbb{R} \geq 1$, the $L_{p,q}$-norm of $\mathbf{X}$ is defined as*

$$\|\mathbf{X}\|_{p,q} = \left(\sum_{j=1}^{n} \left(\sum_{i=1}^{m} |X_{i,j}|^p \right)^{q/p} \right)^{1/q}. \tag{13}$$

When $p = q = 2$, one can observe that the norm in Definition 2.2 becomes the familiar frobenius norm. For consistency, we will refer to the frobenius norm as the $L_{2,2}$-norm. A useful alternative form of Definition 2.2 is

$$\|\mathbf{X}\|_{p,q} = \left(\sum_{j=1}^{n} \|\mathbf{x}_j\|_p^q \right)^{1/q}, \tag{14}$$

where $\mathbf{x}_j$ are the columns of the matrix $\mathbf{X}$. In other words, the matrix $L_{p,q}$-norm can be thought of as taking the L_q-norm of the vector containing the L_p-norms of the columns of $\mathbf{X}$. Therefore, when $q = 1$, it is reasonable to assume that the same sparsity and density inducing effects of the vector L_p-norms apply here as well.

2.3 Principal Component Analysis

In this section, we will give an overview of principal component analysis (PCA) [19]. Over the past several decades, PCA has been applied to several different areas of study including machine learning [20], signal processing [21, 22], and statistics [23]. We begin our overview

of PCA by formulating it from a statistical point of view. Let $\mathbf{Q} \in \mathcal{S}_{d\times k}$ where $k \leq d$ and let $\mathbf{x} \in \mathbb{R}^d$ be a random vector with some covariance matrix and mean vector. The PCA problem can be written as

$$\arg\max_{\mathbf{Q}\in\mathcal{S}_{d\times k}} \operatorname{Tr}\left(\operatorname{cov}\left(\mathbf{Q}^T\mathbf{x}\right)\right) = \arg\max_{\mathbf{Q}\in\mathcal{S}_{d\times k}} \operatorname{Tr}\left(\mathbf{Q}^T\boldsymbol{\Sigma}\mathbf{Q}\right), \tag{15}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix of $\mathbf{x}$. The formulation in (15) indicates that PCA aims to find a k-dimensional projection that preserves as much variance in the data as possible. For this reason, the columns of $\mathbf{Q}$ are often referred to as the principal directions or components. Since $\boldsymbol{\Sigma}$ is positive semi-definite, it admits an eigenvalue decomposition of $\mathbf{PDP}^T$, and

$$\begin{aligned}
\arg\max_{\mathbf{Q}\in\mathcal{S}_{d\times k}} \operatorname{Tr}\left(\mathbf{Q}^T\boldsymbol{\Sigma}\mathbf{Q}\right) &= \arg\max_{\mathbf{Q}\in\mathcal{S}_{d\times k}} \operatorname{Tr}\left(\mathbf{Q}^T\mathbf{PDP}^T\mathbf{Q}\right) && (16)\\
&= \arg\max_{\mathbf{Q}\in\mathcal{S}_{d\times k}} \operatorname{Tr}\left(\mathbf{P}^T\mathbf{QQ}^T\mathbf{PD}\right) && (17)\\
&= \arg\max_{\mathbf{Q}\in\mathcal{S}_{d\times k}} \sum_{i=1}^{d} \lambda_i \mathbf{p}_i^T\mathbf{QQ}^T\mathbf{p}_i. && (18)
\end{aligned}$$

The matrix $\mathbf{QQ}^T$ is an orthogonal projector onto a k-dimensional subspace of $\mathbb{R}^d$ and $0 \leq \mathbf{p}_i^T\mathbf{QQ}^T\mathbf{p}_i \leq 1$ for any choice of $\mathbf{Q}$. Since we have $\mathbf{p}_i^T\mathbf{QQ}^T\mathbf{p}_i = 1$ if and only if $\mathbf{p}_i$ is in the span of the columns of $\mathbf{Q}$, it is easy to see that a solution to (15) is given when the top k eigenvectors are in the span of the columns of $\mathbf{Q}$. Hence, solutions to PCA are rotationally invariant—that is, any $\mathbf{Q}$ that satisfies $\operatorname{span}(\mathbf{Q}) = \operatorname{span}(\mathbf{P}_{:,1:k})$ will be a solution to (15). However, in most practical use cases, it is sufficient to set $\mathbf{Q} = \mathbf{P}_{:,1:k}$ for convenience.

Now consider a zero-mean data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$ with n samples of dimension d. The estimate of the covariance matrix is $\frac{1}{n}\mathbf{XX}^T$, thus,

$$\arg\max_{\mathbf{Q}\in\mathcal{S}_{d\times k}} \operatorname{Tr}\left(\mathbf{Q}^T\boldsymbol{\Sigma}\mathbf{Q}\right) \approx \arg\max_{\mathbf{Q}\in\mathcal{S}_{d\times k}} \operatorname{Tr}\left(\mathbf{Q}^T\mathbf{XX}^T\mathbf{Q}\right) = \arg\max_{\mathbf{Q}\in\mathcal{S}_{d\times k}} \left\|\mathbf{Q}^T\mathbf{X}\right\|_{2,2}^2. \tag{19}$$

Furthermore, for any $\mathbf{X} \in \mathbb{R}^{d \times n}$, it can be shown that

$$\arg\min_{\mathbf{Q} \in \mathcal{S}_{d \times k}} \left\| \mathbf{X} - \mathbf{Q}\mathbf{Q}^T\mathbf{X} \right\|_{2,2}^2 = \arg\max_{\mathbf{Q} \in \mathcal{S}_{d \times k}} \left\| \mathbf{Q}^T\mathbf{X} \right\|_{2,2}^2 . \tag{20}$$

The left hand side of (20) is the familiar low-rank matrix approximation problem where $\mathbf{Q}\mathbf{Q}^T$ represents a projection matrix onto a low-rank subspace [24]. Since (20) involves the $L_{2,2}$-norm, we will refer to this type of PCA as L2-PCA. Obtaining the eigenvectors of $\mathbf{X}\mathbf{X}^T$ is equivalent to obtaining the left hand singular vectors of $\mathbf{X}$. Thus, a solution to (20) is found whenever $\mathbf{Q}$ satisfies $\text{span}(\mathbf{Q}) = \text{span}(\mathbf{U}_{:,1:k})$, where $\mathbf{X} \overset{\text{SVD}}{\rightarrow} \mathbf{U}\mathbf{S}\mathbf{V}^T$. Again, in most practical scenarios, it is sufficient to set $\mathbf{Q} = \mathbf{U}_{:,1:k}$. The PCA problem formulation in (15) is often used interchangeably with (20) despite the subtle condition that the data matrix must be zero-mean for them to be equivalent. However, in applications such as dimensionality reduction, no harm is done when using a data matrix with a non-zero mean.

The squared emphasis imposed by the formulations of L2-PCA in (20) via the $L_{2,2}$-norm make it sensitive to outlier data points. Depending on the magnitude and orientation of these outlier data points, the principal components can be severely corrupted—even if only a few outliers exist in the data. As a result, researchers have focused their efforts on more robust formulations of PCA that utilize the $L_{1,1}$-norm [25, 26, 27, 28, 29, 30, 31, 31, 32] and $L_{2,1}$-norms [33]. The success of $L_{1,1}$-norm based PCA is attributed to the fact that it promotes dense projection components which help to more evenly distribute the effect of outliers in the data. Other robust methods for PCA include [34, 35]. However, if the $L_{2,2}$-norm is swapped for the $L_{p,q}$-norm in PCA, in general, the equality in (20) no longer holds and the problems become much more difficult to solve. At the time of writing, only special cases of these problems have been solved exactly or approximately. That being said, these robust formulations of PCA achieve better general outlier rejection if the data has a low-rank structure.

From Definition 2.2, we observe that q/p places emphasis on the columns of a matrix,

while p places emphasis on the individual elements of a matrix. This seems to suggest that setting $q = 1$ gives more robustness against outlier samples if the outlier samples corrupt the columns of the matrix under the $L_{p,q}$-norm. While some works provide insight into $p = 1$ [27, 30] and $p = 2$ [33], not much is known about the usage of $p > 2$ in PCA. We hope to provide some insight on this matter in this work.

2.4 Subspace Clustering

2.4.1 Overview

In this section, we will discuss the subspace clustering problem and some common algorithms that exist in the literature. Subspace clustering techniques have been successfully applied to real world problems such as face clustering [36], image segmentation [37], and motion segmentation [38]. Consider a data matrix $\mathbf{X} \in \mathbb{R}^{D \times N}$ containing N samples of dimension D. In subspace clustering, we assume that each $\mathbf{x}_i$ belongs to an unknown union of affine subspaces $\{\mathcal{A}_i\}_{i=1}^{c}$. Subspace clustering algorithms typically aim to find the parameters of each affine subspace in $\{\mathcal{A}_i\}_{i=1}^{c}$ and the affine subspace memberships of each sample $\mathbf{x}_i$. In particular, if $\mathbf{x} \in \mathcal{A}_i$, then there exists a $\mathbf{y} \in \mathbb{R}^{d_i}$ such that $\mathbf{x} = \mathbf{U}_i\mathbf{y} + \mathbf{m}_i$, where $\mathbf{U}_i \in \mathcal{S}_{D \times d_i}$ and $\mathbf{m}_i \in \mathbb{R}^D$. Since an affine subspace can be thought of as a translated linear subspace, the matrix $\mathbf{U}_i$ can be interpreted as an orthogonal basis for the corresponding linear subspace with $\mathbf{m}_i$ being the translation vector. Accordingly, the set $\{d_i\}_{i=1}^{c}$ corresponds to the dimension of each affine subspace.

There are several different approaches to solving the subspace clustering problem. One approach is to use matrix factorizations as in the algorithms of Boult and Brown [1], Costeira and Kanade [2], and Gear [3]. While these methods are simple to implement, they require that the subspaces are linear and independent. This often results in poor performance on real world datasets since these conditions are often violated. In addition, the thresholding process required by such algorithms has been shown to be very sensitive to noise [39, 40]. Algebraic techniques such as GPCA [4] can also be used. However, the main drawback of

GPCA is its high computational complexity. Another approach to subspace clustering is to use statistical based methods. Some examples of such algorithms are PPCA [5], ALC [6], and RANSAC [7]. Lastly, there are spectral clustering [8] based methods. We will discuss both spectral clustering and its applications to subspace clustering in the following sections since it plays an important role in our work. For a more in-depth survey and comparison of subspace clustering algorithms, we direct the reader to the work of Vidal [14].

2.4.2 Spectral Clustering

The spectral clustering algorithm described in the work of Ng et al. [8] takes as input an affinity matrix and the desired number of clusters, and returns the segmentation of the data. An affinity matrix is a symmetric $N \times N$ matrix with positive entries, which we will denote by $\mathbf{W}$. Each element of $\mathbf{W}$ represents a measure of similarity between two samples in our data matrix $\mathbf{X}$, i.e., $W_{i,j} > 0$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar, and $W_{i,j} = 0$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are dissimilar for all $(i,\, j) \in \{1, 2, \ldots, N\}^2$. The greater the value of $W_{i,j}$, the more similar the two points $\mathbf{x}_i$ and $\mathbf{x}_i$ are to each other. An element of the affinity matrix can also be thought of as the weight of an edge that connects two vertices in a graph. This interpretation gives way to the normalized cuts algorithm presented in [41], which is closely related to the spectral clustering algorithm in [8]. Both of these algorithms rely on the mathematical results of spectral graph theory. For further reading on spectral graph theory, see the work of Chung [42].

Algorithm 1 shows the spectral clustering algorithm described in Ng et al. [8]. The procedure first forms the diagonal matrix $\mathbf{D}$ whose entries contain the row sums of $\mathbf{W}$. Next, the normalized graph laplacian $\mathbf{L}$ is formed [42]. The bottom c eigenvectors of $\mathbf{L}$ are then extracted and put into the matrix $\mathbf{P}$. Finally the rows of $\mathbf{P}$ are normalized and fed into the k-means algorithm. The output of the k-means algorithm contains the segmentation of the data. There are many machine learning textbooks that describe the k-means algorithm. One such book is Duda and Hart [43]. We use the Python implementation of k-means from

[44]. At first glance, it is natural to wonder why one wouldn't just perform k-means on the original data matrix. The benefit of spectral clustering lies in the observation that it transforms an affinity into a higher dimensional space where tight centroids are typically formed. This is somewhat reminiscent to how kernels are used in SVM [43]. So spectral clustering may work well if the data does form the traditional centroid like clusters that k-means tends to favor. Of course, it is important to note that good performance depends heavily on the data and how the affinity matrix is formed.

Algorithm 1 Spectral Clustering

Require: $\mathbf{W} = \mathbf{W}^T \in \mathbb{R}^{N \times N}$, $W_{i,j} \geq 0$ for all $(i, j) \in \{1, 2, \ldots, N\}^2$, $c > 1$

procedure SPECTRALCLUSTERING($\mathbf{W}$, c)
  $\mathbf{D} \leftarrow \text{diag}\left(\begin{bmatrix} \sum_{i=1}^{N} W_{1,i} & \sum_{i=1}^{N} W_{2,i} & \ldots & \sum_{i=1}^{N} W_{N,i} \end{bmatrix}\right)$
  $\mathbf{L} \leftarrow \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$
  $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \leftarrow \text{SVD}(\mathbf{L})$
  $\mathbf{P} \leftarrow \mathbf{U}_{:,\mathbf{N-c+1:N}}$
  $\mathbf{N} \leftarrow \text{diag}\left(\begin{bmatrix} \|P_{1,:}\|_2 & \|P_{2,:}\|_2 & \ldots & \|P_{N,:}\|_2 \end{bmatrix}\right)$
  $\mathbf{Y} \leftarrow \mathbf{N}^{-1}\mathbf{P}$
  $\mathbf{y} \leftarrow \text{KMEANS}(\mathbf{Y}, c)$
  return y
end procedure

Now we consider a data matrix $\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \ldots & \mathbf{X}_c \end{bmatrix} \in \mathbb{R}^{D \times N}$, where $N = \sum_{i=1}^{c} N_i$, c is the number of classes, N_i is the number of samples from a particular class, and $\mathbf{X}_i \in \mathbb{R}^{D \times N_i}$ for all $i \in \{1, 2, \ldots, c\}$. Suppose that we have formed an affinity matrix $\mathbf{W}$ from this data and $W_{i,j} = 0$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are not in the same class. Then it is clear that $\mathbf{W}$ is block diagonal, with each block on the diagonal corresponding to a distinct class. Ng et al. [8] show that Algorithm 1 can obtain perfect clustering under these conditions.

2.4.3 Subspace Affinities

Now the question becomes: how does one choose to form the affinity matrix? In general, there is no consensus on this matter since it depends heavily on the structure of the clusters in the data [45]. In the work of Ng et al. [8], the gaussian kernel is used as a similarity metric to form the entries of $\mathbf{W}$. In this work, we will focus on methods for forming affinity matrices for subspace clustering. The gaussian kernel is not suitable for subspace clustering since the distance between two points does not provide any information about subspace membership.

One general approach to forming affinity matrices for subspace clustering is to assume that it is often the case that a point and its nearest neighbors belong to the same subspace. This approach is used in the LSA algorithm described in [11]. For each data sample, the LSA algorithm uses L2-PCA to find the subspace formed by its nearest neighbors. The principal angles between these subspaces are used in a similarity metric to determine the entries of the affinity matrix. One drawback of this approach is that it only can handle linear subspaces.

Another common algorithm used to form affinity matrices for subspace clustering is SSC [9]. The main idea behind the SSC algorithm is to notice that each point can be written as an affine combination of other points in the same affine subspace. For this to be feasible, we must have $N_i \geq d_i + 2$ for $i \in \{1, 2, \ldots, c\}$. This can be written as the following convex optimization problem,

$$\underset{\mathbf{C} \in \mathbb{R}^{N \times N}}{\arg\min} \|\mathbf{C}\|_{1,1} \text{ s.t. } \mathbf{X} = \mathbf{X}\mathbf{C}, \operatorname{diag}(\mathbf{C}) = \mathbf{0}_N, \mathbf{C}\mathbf{1}_N = \mathbf{1}_N. \tag{21}$$

The optimal $\mathbf{C}$ can be found using most convex optimization solvers and the affinity matrix $\mathbf{W}$ is formed by $|\mathbf{C}| + |\mathbf{C}|^T$. The $L_{1,1}$-norm in the objective function of (21) induces sparsity in $\mathbf{C}$ which is essential for good clustering performance since the goal is to write each point only in terms of other points that are on the same affine subspace. In the ideal case, each column of $\mathbf{C}$ should contain $d_i + 1$ non-zero elements which in turn results in a block diagonal $\mathbf{W}$. From the previous section, we saw that spectral clustering will achieve perfect clustering

under these conditions.

The final algorithm that we will discuss is called SCC [10]. Rather than using a 2-way affinity matrix $\mathbf{W}$, SCC first forms a $(d+2)$-way affinity tensor $\mathcal{W} \in \mathbb{R}^{N \times N \times \cdots \times N}$. Each element of $\mathcal{W}$ is related to the volume of the convex hull formed by the corresponding $d+2$ points. This volume is zero when the points are in the same affine subspace and non-zero otherwise. However, these volumes are not used alone as the elements of $\mathcal{W}$. The authors introduce the concept of polar curvature [10], which utilizes the volumes of these convex hulls. The polar curvature is used over the volumes because it is invariant to data transforms. Finally, the authors propose a method for reshaping the affinity tensor into a 2-way affinity matrix so it can be used in spectral clustering. One of the main drawbacks of this algorithm is the size of the affinity tensor and the complexity of the polar curvature procedure. To mitigate some of these drawbacks, the authors in [10] propose an iterative sampling method to partially form the affinity tensor $\mathcal{W}$, which greatly reduces the computation time.

3 Problem Motivation

3.1 Gram Subspace Affinities

Another simple approach to forming subspace affinity matrices is to use a gram matrix. For a given data matrix $\mathbf{X} \in \mathbb{R}^{D \times N}$, the gram matrix $\mathbf{G} \in \mathbb{R}^{N \times N}$ has entries $G_{i,j} = \mathbf{x}_i^T \mathbf{x}_j$. It can also be represented as $\mathbf{G} = \mathbf{X}^T \mathbf{X}$. One can observe that the entries of the gram matrix relate to the angle between the corresponding data points. Hence, for the gram matrix to work as a subspace affinity matrix, we must have $\mathbf{x}_i^T \mathbf{x}_j = 0$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are not in the same class. It follows that an equivalent statement would be to require that the classes form orthogonal linear subspaces of $\mathbb{R}^D$. We will now expand on this idea in more detail.

Consider data from c nominal clusters or classes. Suppose that for all $i \in \{1, 2, \ldots, c\}$, we have $\mathbf{U}_i \in \mathcal{S}_{D \times r_i}$, $\mathbf{m}_i \in \mathbb{R}^D$, $\mathbf{D}_i \in \mathbb{R}^{r_i \times r_i}$, and $\mathbf{Z}_i \in \mathbb{R}^{r_i \times N_i}$, such that N_i data from clusters

i are of the form:

$$\mathbf{X}_i = \mathbf{U}_i \mathbf{D}_i \mathbf{Z}_i + \mathbf{m}_i \mathbf{1}_{N_i}^T \in \mathbb{R}^{D \times N_i}. \tag{22}$$

In addition, we require that the $\mathbf{D}_i$ are diagonal, and that $\mathbb{E}\left([\mathbf{Z}_i]_{:,n} [\mathbf{Z}_i]_{:,n}^T \right) = \mathbf{I}_{r_i}$ and $\mathbb{E}\left([\mathbf{Z}_i]_{:,n} \right) = \mathbf{0}_{r_i}$ for all $n \in \{1, 2, \ldots, N_i\}$. Each column of $\mathbf{X}_i$ is a random vector that belongs the $(r_i + 1)$-dimensional affine subspace defined by the orthogonal basis $\mathbf{U}_i$ and translation vector $\mathbf{m}_i$. Also, recall that we have implicitly required that the columns of each $\mathbf{X}_i$ are uncorrelated to each other with respect to the affine subspace that they lie on. Accordingly, each of the matrices in $\{\mathbf{X}_i\}_{i=1}^c$ represent a subspace clustering or class. Furthermore, it holds that

$$\mathbb{E}\left([\mathbf{X}_i]_{:,n} \right) = \mathbf{m}_i \tag{23}$$

and

$$\mathbb{E}\left([\mathbf{X}_i]_{:,n} [\mathbf{X}_i]^T \right) = \mathbf{U}_i \mathbf{D}^2 \mathbf{U}_i^T + \mathbf{m}_i \mathbf{m}_i^T = \mathbf{T}_i \mathbf{\Delta}_i \mathbf{T}_i^T, \tag{24}$$

where

$$\mathbf{T}_i = \begin{bmatrix} \mathbf{U}_i & \frac{\mathbf{m}_i}{\beta_i} \end{bmatrix}, \tag{25}$$

$$\mathbf{\Delta}_i = \begin{bmatrix} \mathbf{D}_i^2 & \mathbf{0}_{r_i} \\ \mathbf{0}_{r_i}^T & \beta_i^2 \end{bmatrix}, \tag{26}$$

and $\beta_i = \|\mathbf{m}_i\|_2$. This implies that we can write

$$\mathbf{X}_i = \mathbf{T}_i \mathbf{\Phi}_i, \tag{27}$$

where

$$\mathbf{\Phi}_i = \begin{bmatrix} \mathbf{D}_i & \mathbf{Z}_i \\ \beta_i & \mathbf{1}_{N_i}^T \end{bmatrix}. \tag{28}$$

If $N_i \geq r_i + 1$, then it can be seen that (25) will tend to have linearly independent columns.

This observation coupled with (27) implies that $\text{span}\,(\mathbf{T}_i) = \text{span}\,(\mathbf{X}_i)$ and $\text{rank}\,(\mathbf{X}_i) = \text{rank}\,(\mathbf{T}_i) \leq r_i + 1$. Now we consider the full data matrix

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \dots & \mathbf{X}_c \end{bmatrix} = \mathbf{T}\mathbf{\Phi} \in \mathbb{R}^{D \times N}, \tag{29}$$

where

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_1 & \mathbf{T}_2 & \dots & \mathbf{T}_c \end{bmatrix}, \tag{30}$$

$$\mathbf{\Phi} = \text{diag}\,(\mathbf{\Phi}_1, \mathbf{\Phi}_2, \dots, \mathbf{\Phi}_c), \tag{31}$$

and $N = \sum_{m=1}^{c} N_m$. The gram matrix formed by the columns of $\mathbf{X}$ is

$$\mathbf{G} = \mathbf{X}^T\mathbf{X} = \mathbf{\Phi}^T\mathbf{T}^T\mathbf{T}\mathbf{\Phi}, \tag{32}$$

which leads us to Proposition 3.1.

Proposition 3.1. *Consider data in c classes that follows the model of (22) and (29). If* $\mathbf{T}^T\mathbf{T} = \mathbf{I}_N$*, then* $\mathbf{G} = \mathbf{\Phi}^T\mathbf{\Phi} = \text{diag}\left(\mathbf{\Phi}_1^T\mathbf{\Phi}_1, \mathbf{\Phi}_2^T\mathbf{\Phi}_2, \dots, \mathbf{\Phi}_c^T\mathbf{\Phi}_c\right)$ *is block diagonal. In addition, the affinity matrix* $\mathbf{W}$ *with* $W_{i,j} = |G_{i,j}|$ *is also block diagonal and will be able to achieve a perfect segmentation of the data via spectral clustering [8]. Furthermore, it can be shown that* $\mathbf{T}^T\mathbf{T} = \mathbf{I}_N \iff \mathbf{U}_i^T\mathbf{U}_j = \mathbf{0}\,\forall i \neq j$, $\mathbf{U}_i^T\mathbf{m}_j = \mathbf{0}\,\forall i, j$, *and* $\mathbf{U}_i^T\mathbf{U}_i = \mathbf{I}_{r_i}\,\forall i$.

Assuming that we now have $\mathbf{T}^T\mathbf{T} = \mathbf{I}_N$, we form the block diagonal matrix $\mathbf{Y} = \mathbf{T}^T\mathbf{X} = \mathbf{\Phi}$. Some observations about $\mathbf{Y}$ are now in order:

i. The column vector $\mathbf{y}_i$ is sparse with $\|y\|_0 \leq \max_\ell (r_\ell + 1)$ for all i. The $\|\cdot\|_0$ quasi-norm denotes the number of non-zero elements of its argument.

ii. The sparsity pattern in $\mathbf{y}_i$ identifies uniquely the cluster of $\mathbf{x}_i$. We call will call this property **cluster-ID sparsity**.

iii. If $\mathbf{x}_i$ and $\mathbf{x}_j$ come from different clusters, then $\mathbf{y}_i^T \mathbf{y}_j = 0$.

The above statements also hold true for the matrix $\mathbf{A}$ where $A_{i,j} = |Y_{i,j}|$ and remain valid even after row or column normalization is applied. This is a result of the sparsity and block diagonal structure of $\mathbf{Y}$.

3.2 Orthogonality in High-Dimensions and Cluster-ID Sparsity

Proposition 3.1 now motivates us to consider scenarios where it is reasonable to assume that $\mathbf{T}^T\mathbf{T} = \mathbf{I}_N$. First, we will consider independently drawn points in $\mathbb{R}^D$. Consider a set of independently drawn points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ where $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}_D, \mathbf{I}_D)$ for all $i \in \{1, 2, \ldots, N\}$. The (i, j)-th entry of the heatmaps in Fig. 3 correspond to the cosine similarity between the two points $\mathbf{x}_i$ and $\mathbf{x}_j$ for one realization of $N = 50$ points. The cosine similarity is defined as

$$s_{\cos}(\mathbf{x}, \mathbf{y}) = \frac{\left|\mathbf{x}^T\mathbf{y}\right|}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2}, \tag{33}$$

and is equal to 0 when $\mathbf{x}$ and $\mathbf{y}$ are orthogonal, and 1 when $\mathbf{x}$ and $\mathbf{y}$ are collinear.

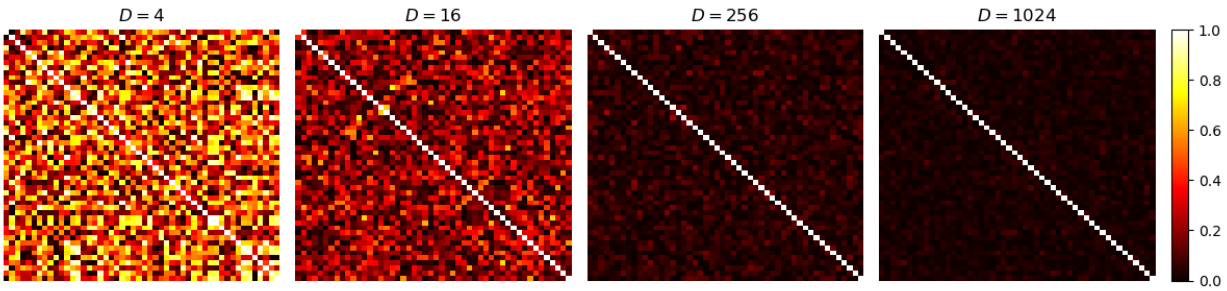

Figure 3: Independently drawn random vectors tend to be orthogonal as D increases. The (i, j)-th entry of each heatmap shows the cosine similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. Here we have $N = 50$ points and $(i, j) \in \{1, 2, \ldots, N\}^2$.

Fig. 3 suggests that zero-mean independent and identically distributed (iid) points tend to be orthogonal as D increases asymptotically. In a similar manner, independently drawn subspaces defined by a basis $\mathbf{U}_i \in \mathcal{S}_{D \times r_i}$ can be compared. For any two matrices $\mathbf{Q}_1 \in \mathcal{S}_{D \times r_1}$

and $\mathbf{Q}_2 \in \mathcal{S}_{D\times r_2}$ with $r_1, r_2 \leq D$, we define the normalized subspace distance as

$$d_{\text{nsub}}\left(\mathbf{Q}_1, \mathbf{Q}_2\right) = \frac{\left\|\mathbf{Q}_1\mathbf{Q}_1^T - \mathbf{Q}_2\mathbf{Q}_2^T\right\|_{2,2}^2}{r_1 + r_2}. \tag{34}$$

Expanding the frobenius norm in (34) yields

$$d_{\text{nsub}}\left(\mathbf{Q}_1, \mathbf{Q}_2\right) = \frac{\left\|\mathbf{Q}_1\mathbf{Q}_1^T - \mathbf{Q}_2\mathbf{Q}_2^T\right\|_{2,2}^2}{r_1 + r_2} \tag{35}$$

$$= \frac{r_1 + r_2 - 2\left\|\mathbf{Q}_1^T\mathbf{Q}_2\right\|_{2,2}^2}{r_1 + r_2}, \tag{36}$$

and it becomes clear that

$$\frac{r_1 + r_2 - 2\min\left(r_1, r_2\right)}{r_1 + r_2} \leq d_{\text{nsub}}\left(\mathbf{Q}_1, \mathbf{Q}_2\right) \leq 1. \tag{37}$$

Furthermore, we note that $r_1 + r_2 \leq D$ is a necessary condition for (34) to be 0. Conversely, $r_1 = r_2$ is a necessary condition for (34) to be 1. When (34) is 1, the subspaces are orthogonal. The subspaces are identical if (34) is 0. Fig. 4 shows the average normalized subspace error between $\mathbf{Q}_1, \mathbf{Q}_1 \in \mathcal{S}_{D\times r}$, where

$$\mathbf{Q}_i \in \mathcal{S}_{D\times n} \text{ s.t. } \mathbf{R}_i \overset{\text{SVD}}{\rightarrow} \mathbf{Q}_i\mathbf{\Sigma}_i\mathbf{V}_i^T \text{ and } \left[\mathbf{R}_i\right]_{:,j} \sim \mathcal{N}\left(\mathbf{0}_D, \mathbf{I}_D\right), \tag{38}$$

for all $i \in \{1, 2\}$ and $j \in \{1, 2, \ldots, r\}$. Each point in Fig. 4 is the average of 10000 realizations with $r = 8$.

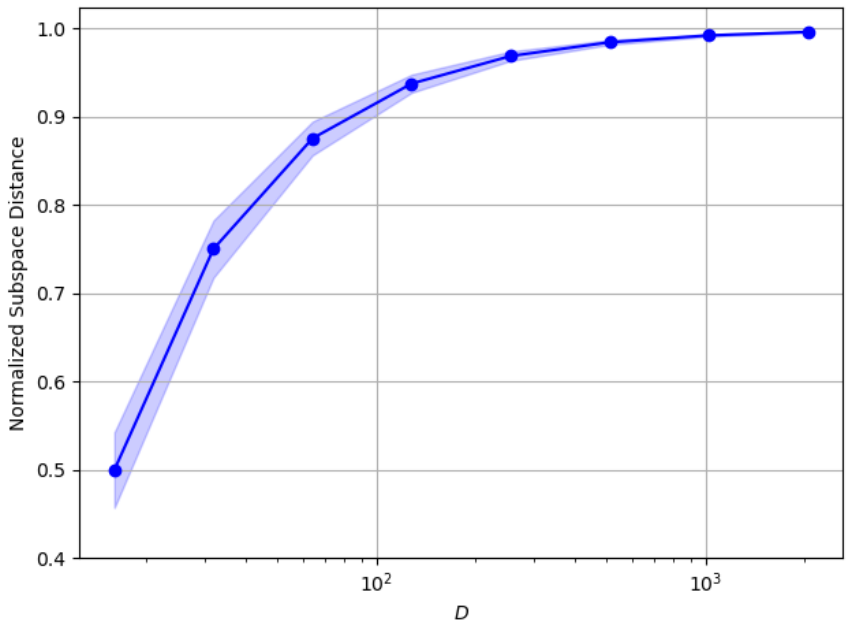

Figure 4: We show the average normalized subspace error versus D. Independent and identically distributed low-rank linear subspaces tend to be orthogonal as D increases. The rank of both subspaces is $r = 8$.

Fig. 4 suggests that as $r = 8 << D$, on average, iid subspaces following the distribution of (38) tend to be orthogonal.

Fig. 3 and Fig. 4 seem to suggest that $\mathbf{T}^T\mathbf{T} = \mathbf{I}_N$ might be a viable assumption if the rank of the data is much smaller than the ambient dimension D. Fig. 5 plots the average squared error $\left\|\mathbf{I}_N - \mathbf{T}^T\mathbf{T}\right\|_{2,2}^2$ versus D. Each point in Fig. 5 is the average of 10000 realizations. For each of these realizations, (25) is used to form $\mathbf{T} = \begin{bmatrix}\mathbf{T}_1 & \mathbf{T}_2\end{bmatrix}$ using iid drawn subspaces $\mathbf{U}_1, \mathbf{U}_1 \in \mathcal{S}_{D\times r}$ via (38) and $\mathbf{m}_1, \mathbf{m}_2 \in \mathbb{R}^D$ that are drawn from the standard normal distribution with $r = 4$.

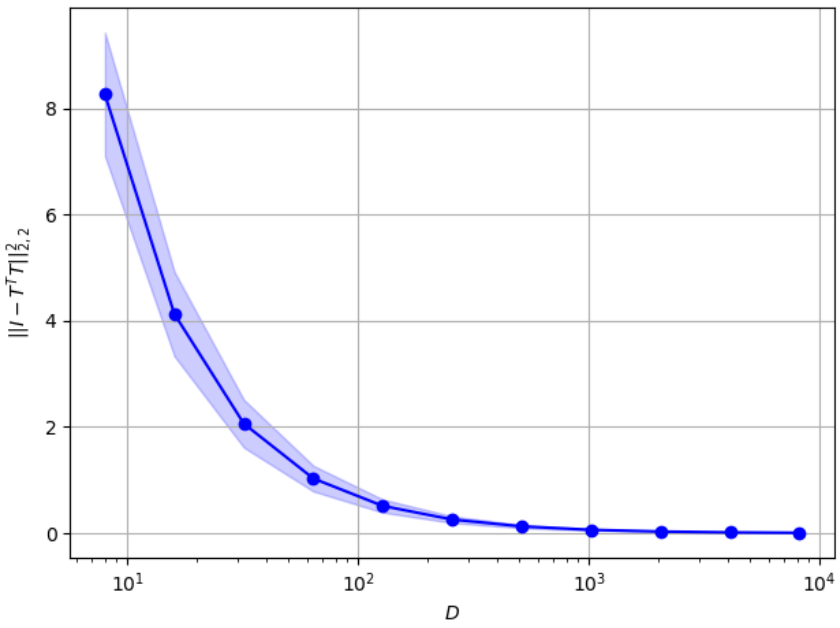

Figure 5: Here we have plotted the average squared error $\left\|\mathbf{I} - \mathbf{T}^T\mathbf{T}\right\|_{2,2}^2$ versus D with $r = 4$. The matrix $\mathbf{T}$ from (25) becomes orthogonal as D increases.

The results of Fig. 5 suggest that $\mathbf{T}^T\mathbf{T}$ is a viable assumption when given data that follows the model in (22) and (29), and $\sum_{i=1}^{c} r_i + 1 << D$.

In practice, there are many datasets that satisfy our requirement of $\sum_{i=1}^{c} r_i + 1 << D$. Some general categories of datasets that have high dimensionality and low-rank structure include images, videos, and genomic data. When the clusters form distinct orthogonal subspaces, Proposition 3.1 implies that the subspace affinity matrix formed by taking the gram matrix of the data matrix $\mathbf{X}$ will result in perfect clustering. However, it is unlikely that real data follow the distributions in (22) and (29). Furthermore, the affine subspaces representing each cluster in the data will likely be approximately orthogonal to one another. Under such non-ideal circumstances, one approach is to a find some projection $\mathbf{Q} \in \mathcal{S}_{D\times k}$ onto the data such that the matrix $\mathbf{Y} = \mathbf{Q}^T\mathbf{X}$ has cluster-ID sparsity. If cluster-ID sparsity is attained, then the subspace affinity matrix formed via the gram matrix of $\mathbf{Y}$ will result in perfect clustering. Using standard PCA to find such a $\mathbf{Q}$ will generally not result in a $\mathbf{Y}$ that has cluster-ID sparsity since the L_2-norm does not promote sparsity. We therefore seek at type of PCA that promotes sparsity in the columns of $\mathbf{Y}$.

4 Proposed LP1-PCA for Low-Rank Clustering

4.1 Formulation

In response to the discussion in the previous section, we propose a novel sparsity inducing PCA called LP1-PCA that promotes cluster-ID sparsity. Consider a data matrix $\mathbf{X} \in \mathbb{R}^{D \times N}$ that contains N samples of dimension D. Let $p \geq 1$ and $k \leq \min(D, N)$. The LP1-PCA problem is defined as

$$\max_{\mathbf{Q} \in \mathcal{S}_{D \times k}} \left\| \mathbf{Q}^T \mathbf{X} \right\|_{p,1} = \max_{\mathbf{Q} \in \mathcal{S}_{D \times k}} \sum_{i=1}^{N} \left\| \mathbf{Q}^T \mathbf{x}_i \right\|_p . \tag{39}$$

Remark. *The LP1-PCA problem in (39) is NP-hard.*

Proof. First, we begin by considering a special case of (39) where $p = 1$ and $k = 1$. The problem in (39) now becomes the L1-PCA problem discussed in the work of [27] who show that the L1-PCA problem reduces to the NP-complete equal-partition problem [46]. Thus, we can conclude that the general LP1-PCA problem in (39) is NP-hard. □

The formulation of LP1-PCA in (39) aims to find a projection $\mathbf{Q}$ onto the data matrix $\mathbf{X}$ that maximizes the $L_{p,1}$-norm. For our purposes, we are interested in $p > 2$, since it will induce sparsity in the columns of $\mathbf{Q}^T\mathbf{X}$ which helps enforce cluster-ID sparsity. The resulting $\mathbf{Q}^T\mathbf{X}$ can be used to form gram matrix affinities that can be used in spectral clustering.

4.2 Exact Solver for Rank-1 Analysis

Now we will derive an exact solution to (39) when $\text{rank}(\mathbf{X}) = 1$. We will also show that the solution in this special case will yield meaningful insight into the selection of p in LP1-PCA. Since $\text{rank}(\mathbf{X}) = 1$, we know that $\mathbf{X} \overset{\text{SVD}}{\rightarrow} \mathbf{u}\sigma\mathbf{v}^T$. Thus, the columns of the data matrix $\mathbf{X}$

can be rewritten as scalar multiples of $\mathbf{u}$, i.e.,

$$\sum_{i=1}^{N} \left\|\mathbf{Q}^T\mathbf{x}_i\right\|_p = \sum_{i=1}^{N} \left\|\mathbf{Q}^T\sigma v_i\mathbf{u}\right\|_p \tag{40}$$

$$= \left(\sum_{i=1}^{N} |\sigma v_i|\right) \left\|\mathbf{Q}^T\mathbf{u}\right\|_p . \tag{41}$$

Given this result, rather than solving (39), it suffices to solve

$$\max_{\mathbf{Q}\in\mathcal{S}_{D\times k}} \left\|\mathbf{Q}^T\mathbf{u}\right\|_p . \tag{42}$$

Define $q = p/(p-1)$ and let $\mathbf{b} \in \mathbb{R}^k$. By Corollary 2.1.3, we have

$$\max_{\mathbf{Q}\in\mathcal{S}_{D\times k}} \left\|\mathbf{Q}^T\mathbf{u}\right\|_p = \max_{\mathbf{Q}\in\mathcal{S}_{D\times k}} \max_{\|\mathbf{b}\|_q\leq 1} \mathbf{b}^T\mathbf{Q}^T\mathbf{u} \tag{43}$$

$$= \max_{\|\mathbf{b}\|_q\leq 1} \max_{\mathbf{Q}\in\mathcal{S}_{D\times k}} \mathbf{b}^T\mathbf{Q}^T\mathbf{u} \tag{44}$$

$$= \max_{\|\mathbf{b}\|_q\leq 1} \|\mathbf{b}\|_2 . \tag{45}$$

Since $\left\|\mathbf{Q}^T\mathbf{u}\right\|_2 = 1$, equality in (45) is achieved whenever $\mathbf{Q}^T\mathbf{u} = \mathbf{b}/\|\mathbf{b}\|_2$.

Lemma 4.1. *Suppose that* $\mathbf{b}^*$ *solves (45), then*

$$\mathbf{Q}^* = \begin{bmatrix}\mathbf{u} & \tilde{\mathbf{U}}_{D\times k-1}\end{bmatrix} \begin{bmatrix}\mathbf{p}^T \\ \tilde{\mathbf{P}}^T_{k\times k-1}\end{bmatrix} = \mathbf{u}\mathbf{p}^T + \tilde{\mathbf{U}}\tilde{\mathbf{P}}^T \tag{46}$$

solves (42), where

$$\mathbf{u}\mathbf{b}^{*T} \overset{SVD}{\rightarrow} \begin{bmatrix}\mathbf{u} & \tilde{\mathbf{U}}_{D\times k-1}\end{bmatrix} \operatorname{diag}\left(\begin{bmatrix}\|\mathbf{b}^*\|_2 & \mathbf{0}^T_{k-1}\end{bmatrix}\right) \begin{bmatrix}\mathbf{p}^T \\ \tilde{\mathbf{P}}^T_{k\times k-1}\end{bmatrix}, \tag{47}$$

$\mathbf{p} = \mathbf{b}^*/\|\mathbf{b}^*\|_2$, $\tilde{\mathbf{U}} \in \mathcal{S}_{D\times k-1}$, $\tilde{\mathbf{P}} \in \mathcal{S}_{k\times k-1}$, *and* $\tilde{\mathbf{U}}^T\mathbf{u} = \tilde{\mathbf{P}}^T\mathbf{p} = \mathbf{0}_{k-1}$.

Next, we focus on identifying the $\mathbf{b} \in \mathbb{R}^k$ that solve (45).

Lemma 4.2. *Let $(m, n) \in \{(m, n) \in \mathbb{R}^2 : 1 \leq m \leq n\}$, then for any $\mathbf{x} \in \mathbb{R}^d$ we have*

$$\|\mathbf{x}\|_n \leq \|\mathbf{x}\|_m \leq d^{(1/m-1/n)} \|\mathbf{x}\|_n . \tag{48}$$

Furthermore, let $(i, j) \in \{1, 2, \ldots, d\}^2$ and suppose that $\mathbf{x} \neq \mathbf{0}$. Equality on the right in (48) is achieved when $|x_i| = |x_j|$ for all (i, j). Equality on the left in (48) is achieved if there exists an i such that $|x_i| \neq 0$ and $x_j = 0$ for all $j \neq i$.

Lemma 4.2 gives us a set of solutions to (45), i.e.,

$$\mathbf{b}^* \in \begin{cases} \left\{\mathbf{b} \in \mathbb{R}^k : \exists i \, 0 < |b_i| \leq 1 \wedge b_j = 0 \, \forall i \neq j\right\}, & q < 2 \\ \left\{\mathbf{b} \in \mathbb{R}^k : 0 < \|\mathbf{b}\|_2 \leq 1\right\}, & q = 2 \\ \left\{\mathbf{b} \in \mathbb{R}^k : |b_i| = |b_j| \; \forall i, j \wedge 0 < \|\mathbf{b}\|_q \leq 1\right\}, & q > 2 \end{cases} . \tag{49}$$

Accordingly, the set of $\mathbf{p} = \mathbf{b}^* / \|\mathbf{b}^*\|_2$ now becomes

$$\mathbf{p} \in \begin{cases} \left\{\mathbf{p} \in \mathbb{R}^k : \exists i \; |p_i| = 1 \wedge p_j = 0 \, \forall i \neq j\right\}, & q < 2 \\ \left\{\mathbf{p} \in \mathbb{R}^k : \|\mathbf{p}\|_2 = 1\right\}, & q = 2 \\ \left\{\pm k^{-1/2}\right\}^k, & q > 2 \end{cases} . \tag{50}$$

Thus, the result of Lemmas 4.1 and 4.2 give us a set of closed form solutions to (39) when $\text{rank}\,(\mathbf{X}) = 1$ that can be obtained by first selecting any $\mathbf{p}$ according to (50) and then calculating $\mathbf{Q}^*$ via the steps in Lemma 4.1.

Now we make an important observation. Consider that we have calculated some $\mathbf{Q}^*$ in

the solution set of (39) for the rank $(\mathbf{X}) = 1$ case, then

$$\mathbf{Q}^{*T}\mathbf{X} = \begin{bmatrix} \mathbf{Q}^{*T}\mathbf{x}_1 & \mathbf{Q}^{*T}\mathbf{x}_2 & \dots & \mathbf{Q}^{*T}\mathbf{x}_n \end{bmatrix} \tag{51}$$

$$= \begin{bmatrix} \mathbf{Q}^{*T}\mathbf{u}\sigma v_1 & \mathbf{Q}^{*T}\mathbf{u}\sigma v_2 & \dots & \mathbf{Q}^{*T}\mathbf{u}\sigma v_n \end{bmatrix} \tag{52}$$

$$= \begin{bmatrix} \mathbf{p}\sigma v_1 & \mathbf{p}\sigma v_2 & \dots & \mathbf{p}\sigma v_n \end{bmatrix}. \tag{53}$$

The columns of (53) represent the projection components of each sample $\mathbf{x}_i$. Considering again (50), we can clearly see that there are three distinct scenarios, each with their own respective solution set. First, when $1 \leq p < 2$, we have $q = p/(p-1) > 2$, thus the projection components of each sample in (53) are dense. Secondly, when $2 < p < \infty$, we have $q = p/(p-1) < 2$, thus the projection components of each sample in (53) are sparse. Finally, when $p = q = 2$, any $\mathbf{p}$ satisfying $\|\mathbf{p}\|_2 = 1$ will suffice, so the projection components of each sample in (53) can range from sparse to dense.

4.3 Approximate Solver for General Rank

We will now present an approximate solution to LP1-PCA in (39) using alternating optimization for the general case of $1 \leq \text{rank}\,(\mathbf{X}) \leq D$. Let $\mathcal{B}^q_{k\times N} = \left\{\mathbf{B} \in \mathbb{R}^{k\times N} : \|\mathbf{b}_i\|_q \leq 1 \,\forall i\right\}$ and $q = p/(p-1)$. By Corollary 2.1.3, we have

$$\max_{\mathbf{Q}\in\mathcal{S}_{D\times k}} \sum_{i=1}^{N} \left\|\mathbf{Q}^T\mathbf{x}_i\right\|_p = \max_{\mathbf{Q}\in\mathcal{S}_{D\times k}} \max_{\mathbf{B}\in\mathcal{B}^q_{k\times N}} \sum_{i=1}^{N} \mathbf{b}_i^T\mathbf{Q}^T\mathbf{x}_i \tag{54}$$

$$= \max_{\mathbf{Q}\in\mathcal{S}_{D\times k}} \max_{\mathbf{B}\in\mathcal{B}^q_{k\times N}} \text{Tr}\left(\mathbf{B}^T\mathbf{Q}^T\mathbf{X}\right) \tag{55}$$

$$= \max_{\mathbf{B}\in\mathcal{B}^q_{k\times N}} \max_{\mathbf{Q}\in\mathcal{S}_{D\times k}} \text{Tr}\left(\mathbf{B}^T\mathbf{Q}^T\mathbf{X}\right) \tag{56}$$

$$= \max_{\mathbf{B}\in\mathcal{B}^q_{k\times N}} \max_{\mathbf{Q}\in\mathcal{S}_{D\times k}} \text{Tr}\left(\mathbf{Q}^T\mathbf{X}\mathbf{B}^T\right) \tag{57}$$

$$= \max_{\mathbf{B}\in\mathcal{B}^q_{k\times N}} \left\|\mathbf{X}\mathbf{B}^T\right\|_*. \tag{58}$$

Equality in (58) is achieved for any $\mathbf{Q}$ that solves the inner maximization problem in (57). However, unlike the $\text{rank}\,(\mathbf{X}) = 1$ case, the problem in (58) is difficult to solve.

Lemma 4.3. *Suppose that* $\mathbf{B}^*$ *solves the outer maximization problem in (57). By Corollary 2.1.3,* $\mathbf{B}^*_{:,i} = \mathbf{V}_{:,i} / \left\|\mathbf{V}_{:,i}\right\|_q$ *and* $\mathbf{V} = \text{sgn}\left(\mathbf{Q}^T\mathbf{X}\right) \circ \left|\mathbf{Q}^T\mathbf{X}\right|^{\circ(p-1)}$ *for all* i*. This results in* $\text{Tr}\left(\mathbf{B}^{*T}\mathbf{Q}^T\mathbf{X}\right) = \sum_{i=1}^N \left\|\mathbf{Q}^T\mathbf{x}_i\right\|_p$.

Lemma 4.4. *Suppose that* $\mathbf{Q}^*$ *solves the inner maximization problem in (57). For a fixed* $\mathbf{B}$*, (57) becomes the orthogonal procrustes problem, and* $\mathbf{Q}^* = \mathbf{U}\mathbf{V}^T$ *with* $\mathbf{X}\mathbf{B}^T \overset{SVD}{\rightarrow} \mathbf{U}_{D\times k}\mathbf{\Sigma}_{k\times k}\mathbf{V}^T_{k\times k}$ *[47].*

Lemmas 4.3 and 4.4 suggest that we can increase the objective function in (57) by alternating between solving the inner and outer maximization problems until convergence. We therefore present Algorithm 2. The first step in Algorithm 2 involves generating an arbitrary $\mathbf{Q} \in \mathcal{S}_{D\times k}$ as initialization. Next, $\mathbf{Q}$ is fixed and $\mathbf{B}$ is set according to Lemma 4.3. Next, $\mathbf{B}$ is fixed and $\mathbf{Q}$ is set according to Lemma 4.4. This process of alternating between the outer and inner maximization of (57) is repeated until the objective function converges in terms of the metric in (39). We should also note that when $p = 1$, Algorithm 2 is equivalent to [30]. Additionally, when $p = 2$, Algorithm 2 is equivalent to [33].

Algorithm 2 LP1-PCA

Require: $p \geq 1$, $\mathbf{Q} \in \mathcal{S}_{D\times k}$, $\mathbf{X} \in \mathbb{R}^{D\times N}$, $1 \leq k \leq D$.

procedure LP1PCA(p, $\mathbf{Q}$, $\mathbf{X}$)
  $q \leftarrow p/(p-1)$
  loop
    $\mathbf{V} \leftarrow \operatorname{sgn}\left(\mathbf{Q}^T\mathbf{X}\right) \circ \left|\mathbf{Q}^T\mathbf{X}\right|^{\circ(p-1)}$
    $\mathbf{N} \leftarrow \operatorname{diag}\left(\begin{bmatrix} \|V_{:,1}\|_q & \|V_{:,2}\|_q & \dots & \|V_{:,N}\|_q \end{bmatrix}\right)$
    $\mathbf{B} = \mathbf{V}\mathbf{N}^{-1}$
    $\mathbf{M} \leftarrow \mathbf{X}\mathbf{B}^T$
    if the metric has converged **then**
      break
    end if
    $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \leftarrow \text{SVD}(\mathbf{M})$
    $\mathbf{Q} \leftarrow \mathbf{U}_{:,1:k}\mathbf{V}^T$
  end loop
  return Q
end procedure

Proposition 4.1. *Algorithm 2 increases the objective function in (39) after each iteration count z.*

Proof. Lemma 4.4 tells us that for any fixed $\mathbf{Q}_{z+1}$, the optimal $\mathbf{B}$ is $\mathbf{B}_{z+1}$. Thus, we can deduce that

$$\operatorname{Tr}\left(\mathbf{Q}_{z+1}^T\mathbf{X}\mathbf{B}_z^T\right) \leq \operatorname{Tr}\left(\mathbf{Q}_{z+1}^T\mathbf{X}\mathbf{B}_{z+1}^T\right) = \sum_{i=1}^{n} \left\|\mathbf{Q}_{z+1}^T\mathbf{x}_i\right\|_p. \tag{59}$$

Similarly, Lemma 4.3 tells us that for any fixed $\mathbf{B}_z$, the optimal $\mathbf{Q}$ is $\mathbf{Q}_{z+1}$. Thus, we can deduce that

$$\sum_{i=1}^{n} \left\|\mathbf{Q}_z^T\mathbf{x}_i\right\|_p = \operatorname{Tr}\left(\mathbf{Q}_z^T\mathbf{X}\mathbf{B}_z^T\right) \leq \operatorname{Tr}\left(\mathbf{Q}_{z+1}^T\mathbf{X}\mathbf{B}_z^T\right). \tag{60}$$

Putting the two inequalities together yields the desired result of

$$\sum_{i=1}^{n} \left\| \mathbf{Q}_z^T \mathbf{x}_i \right\|_p \leq \sum_{i=1}^{n} \left\| \mathbf{Q}_{z+1}^T \mathbf{x}_i \right\|_p . \tag{61}$$

□

Although Proposition 4.1 proves that Algorithm 2 increases the objective function in (39), it may not converge to a global optima, nor is it guaranteed to converge with respect to the argument. Therefore, the performance of Algorithm 2 is dependant on the initialization of $\mathbf{Q}$. To mitigate these drawbacks, we recommend running the algorithm multiple times with different initializations and choosing the solution that has the maximum value of the metric in (39).

Fig. 6 shows the convergence of Algorithm 2 with a single arbitrarily initialized $\mathbf{Q}$ on a synthetic data matrix $\mathbf{X} \in \mathbb{R}^{20\times 100}$ with $\text{rank}\,(\mathbf{X}) = 8$ and $k = 4$ for different values of p.

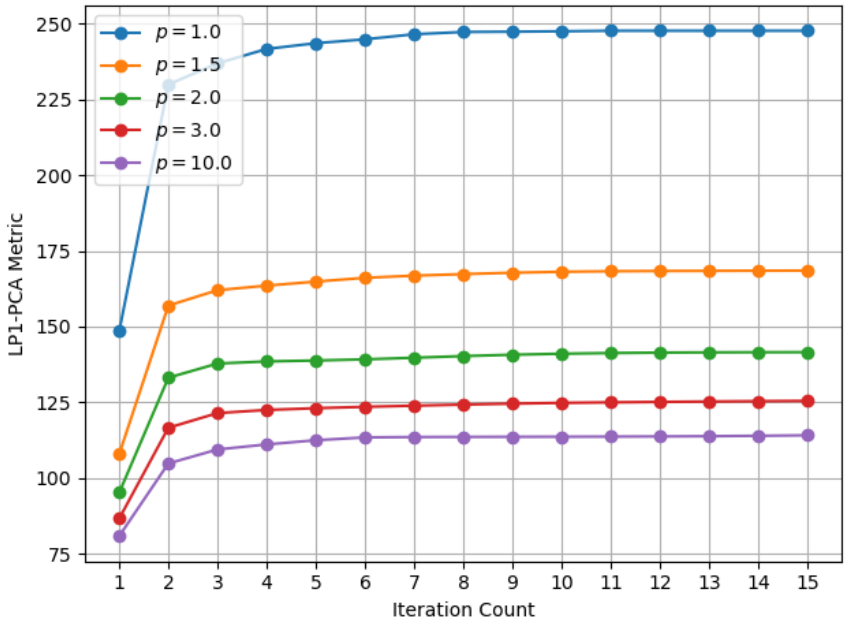

Figure 6: Convergence of Algorithm 2 on a synthetic data matrix $\mathbf{X} \in \mathbb{R}^{20\times 100}$ with $\text{rank}\,(\mathbf{X}) = 8$ and $k = 4$.

Fig. 6 verifies that Algorithm 2 increases the metric in (39) after each iteration. In this case, convergence occurs after about 9 iterations for most values of p.

4.4 LP1-PCA Spectral Clustering

In this section, we propose an algorithm for subspace clustering that utilizes LP1-PCA to produce subspace affinities. We call this algorithm LP1-PCA Spectral Clustering. Algorithm 3 defines the proposed algorithm. We have utilized LP1-PCA as defined in Algorithm 2 and spectral clustering as defined in Algorithm 1.

Algorithm 3 LP1-PCA Spectral Clustering

Require: $p > 2$, $\mathbf{Q} \in \mathcal{S}_{D\times k}$, $\mathbf{X} \in \mathbb{R}^{D\times N}$, $1 \leq k \leq D$, $c > 1$.

1: **procedure** LP1SPECTRAL(p, $\mathbf{Q}$, $\mathbf{X}$, c)

2: $\quad \mathbf{Q} \leftarrow \text{LP1PCA}(p, \mathbf{Q}, \mathbf{X})$

3: $\quad \mathbf{Y} = \mathbf{Q}^T\mathbf{X}$

4: $\quad \mathbf{N} \leftarrow \text{diag}\left(\begin{bmatrix} \|Y_{1,:}\|_2 & \|Y_{2,:}\|_2 & \dots & \|Y_{k,:}\|_2 \end{bmatrix}\right)$

5: $\quad \mathbf{A} = |\mathbf{N}^{-1}\mathbf{Y}|$

6: $\quad \mathbf{W} = \mathbf{A}^T\mathbf{A}$

7: $\quad \mathbf{y} = \text{SPECTRALCLUSTERING}(W, c)$

8: $\quad$ **return y**

9: **end procedure**

Algorithm 3 takes as its arguments an arbitrary $\mathbf{Q} \in \mathcal{S}_{D\times k}$ to be used in LP1-PCA, a data matrix $\mathbf{X}$, a value of p for LP1-PCA, and the number of classes c to cluster. First, Algorithm 3 uses LP1-PCA to find a $\mathbf{Q} \in \mathcal{S}_{D\times k}$ that forms a sparse $\mathbf{Q}^T\mathbf{X}$. In order to achieve such sparsity, we require that $p > 2$. Next an affinity matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$ is formed by computing the gram matrix of $\mathbf{A}$, where the elements of $\mathbf{A}$ are the absolute elements of a row normalized version of $\mathbf{Y} = \mathbf{Q}^T\mathbf{X}$. Finally, the resulting affinity matrix $\mathbf{W}$ and the number of clusters c is using in spectral clustering which returns the segmentation of the data.

The sparsity induced in the columns of $\mathbf{Y}$ as a result of setting $p > 2$ helps to enforce cluster-ID sparsity which is needed to create subspace affinities that will perform well in

spectral clustering. The row normalization and absolute value procedures do not affect cluster-ID sparsity.

5 Experiments

5.1 LP1-PCA Toy Examples

In this section, we explore how p affects the solutions of Algorithm 2 and the sparsity of the matrix $\mathbf{Q}^T\mathbf{X}$. First, we will explore some toy examples in $\mathbb{R}^3$. Fig. 7 shows Algorithm 2 applied to a synthetic rank one data matrix $\mathbf{X} \in \mathbb{R}^{3\times 40}$ with $k = 2$. Unless otherwise stated, all experiments in this section use an arbitrary initialization of $\mathbf{Q}$ for Algorithm 2. Notice that the respective solutions shown in Fig. 7 for $p \in \{1, 2, 10\}$ correspond to our theoretical derivation in (53). Hence, setting $p = 1$ resulted a basis that promotes density in $\mathbf{Q}^T\mathbf{X}$ and $p = 10$ resulted in a basis that promotes sparsity in $\mathbf{Q}^T\mathbf{X}$. In this case, $p = 2$ has arbitrarily chosen a basis that contains the columns of $\mathbf{X}$ in its span.

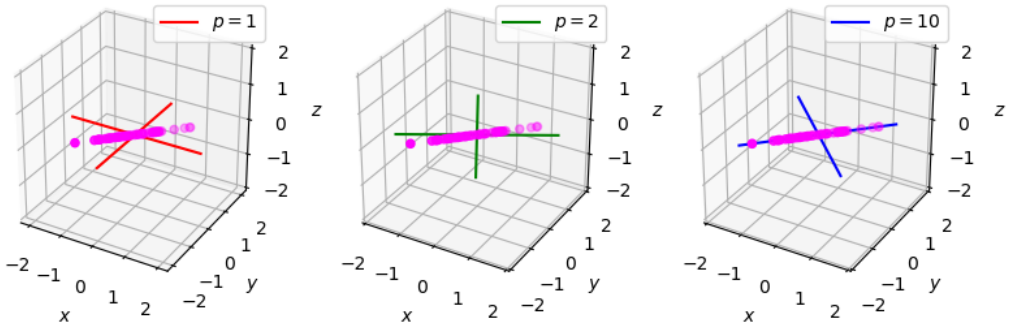

Figure 7: LP1-PCA solutions for a synthetic rank one data matrix $\mathbf{X} \in \mathbb{R}^{3\times 40}$ with $k = 2$.

Fig. 8 shows Algorithm 2 applied to a synthetic rank one data matrix $\mathbf{X} \in \mathbb{R}^{3\times 40}$ with a

small amount of additive white gaussian noise and $k = 2$. Even though the corrupted data matrix is full rank, most of the singular values associated with pure noise are very small, so it can be thought of as approximately rank one. Therefore, $p = 1$ is still able to find a dense solution and $p = 10$ is still able to find a sparse solution. Setting $p = 2$ yielded a solution that is in-between $p = 1$ and $p = 10$ in terms of sparsity.

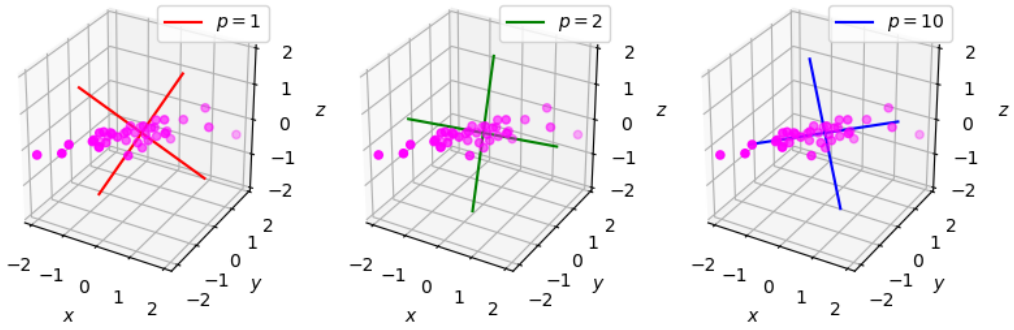

Figure 8: LP1-PCA solutions for a synthetic rank one data matrix $\mathbf{X} \in \mathbb{R}^{3\times 40}$ with a small amount of additive white gaussian noise and $k = 2$

Fig. 9 shows Algorithm 2 applied to a rank two data matrix $\mathbf{X} \in \mathbf{R}^{3\times 40}$ with singular values $\sigma_1 = 1$, $\sigma_2 = 1/2$, and $k = 2$. We note here that the solutions using $p \in \{1, 2, 10\}$ find rotated versions of $\mathbf{Q}$ that form the same subspace spanned by the columns of $\mathbf{X}$. When $p = 1$, the principal directions are oriented to equally capture the direction associated with σ_1. Conversely, when $p = 10$, one principal direction almost completely captures the direction associated with σ_1. In this case, $p = 2$ finds an arbitrarily rotated version of $\mathbf{Q}$ inside of the subspace of the data.

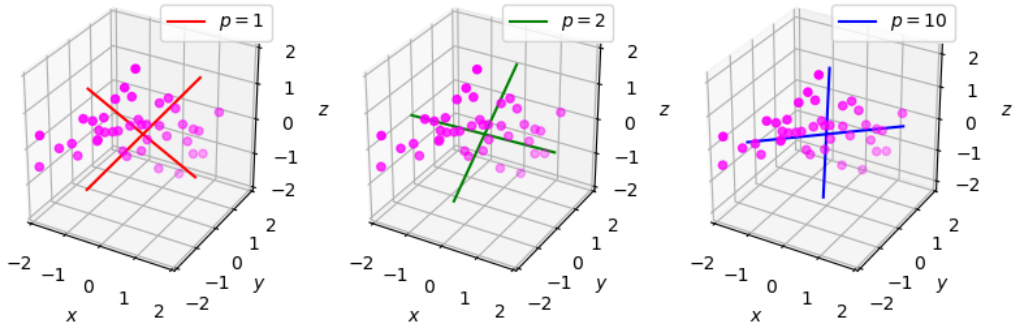

Figure 9: LP1-PCA solutions for a rank two data matrix $\mathbf{X} \in \mathbb{R}^{3\times 50}$ with $k = 2$.

In summary, these results suggest that when $k \geq \text{rank}(\mathbf{X})$, all p will find a $\mathbf{Q}$ that captures the subspace of the data. In this case, the solutions will differ in that $\mathbf{Q}$ will be rotated to induce more sparsity or density depending on the selection of p.

Next, we will quantify the sparsity of $\mathbf{Q}^T\mathbf{X}$ versus p. Consider a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$. Define the set of all elements of $\mathbf{X}$ to be $\mathcal{A}$ and let $\alpha \in [0, 1]$. We can define a measure of density as

$$m_{\text{density}}(\mathbf{X}; \alpha) = \min_{\mathcal{B}} \frac{|\mathcal{B}|}{mn} \text{ s.t. } \mathcal{B} \subseteq \mathcal{A} \text{ and } \sum_{y\in\mathcal{B}} |y| \geq \alpha \sum_{z\in\mathcal{A}} |z|, \tag{62}$$

where $|\mathcal{B}|$ is the cardinality of the set $\mathcal{B}$. In other words, (62) is the cardinality of the smallest subset of elements in $\mathbf{X}$ whose absolute sum is greater than $\alpha \|\mathbf{X}\|_{1,1}$. The later result is divided by the number of elements of $\mathbf{X}$ for normalization. From (62), we can define the sparsity of a matrix $\mathbf{X}$ as

$$m_{\text{sparsity}}(\mathbf{X}; \alpha) = 1 - m_{\text{density}}(\mathbf{X}; \alpha). \tag{63}$$

Equations (62) and (63) are normalized and thus take values from 0 to 1. Unless otherwise

specified, it can be assumed that we set $\alpha = 0.9999$ in (62) and (63). Fig. 10 shows the average density and reconstruction error of $\mathbf{Q}^T\mathbf{X}$ versus p with the measure of density defined in (62) for different values of k. The reconstruction error is defined as

$$e_R\left(\mathbf{X}, \mathbf{Q}\right) = \frac{\left\|\mathbf{X} - \mathbf{Q}\mathbf{Q}^T\mathbf{X}\right\|_{2,2}^2}{\left\|\mathbf{X}\right\|_{2,2}^2}, \tag{64}$$

which quantifies how well the subspace captures the data. Each point in Fig. 10 is the result of applying Algorithm 2 to 1000 iid realizations of $\mathbf{X} \in \mathbb{R}^{500\times 50}$ with $\text{rank}\left(\mathbf{X}\right) = 40$. The matrix $\mathbf{X}$ is formed via $\mathbf{X} = \mathbf{U}\mathbf{Z}$, where

$$\mathbf{U} \in \mathcal{S}_{500\times 40} \text{ s.t. } \mathbf{R} \overset{\text{SVD}}{\rightarrow} \mathbf{Q}\mathbf{\Sigma}\mathbf{V}^T \text{ and } \mathbf{r}_j \sim \mathcal{N}\left(\mathbf{0}_D, \mathbf{I}_D\right), \tag{65}$$

$\mathbf{Z} \in \mathbb{R}^{40\times 50}$, and $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We use the standard L2-PCA solution from (20) to initialize Algorithm 2 in this study.

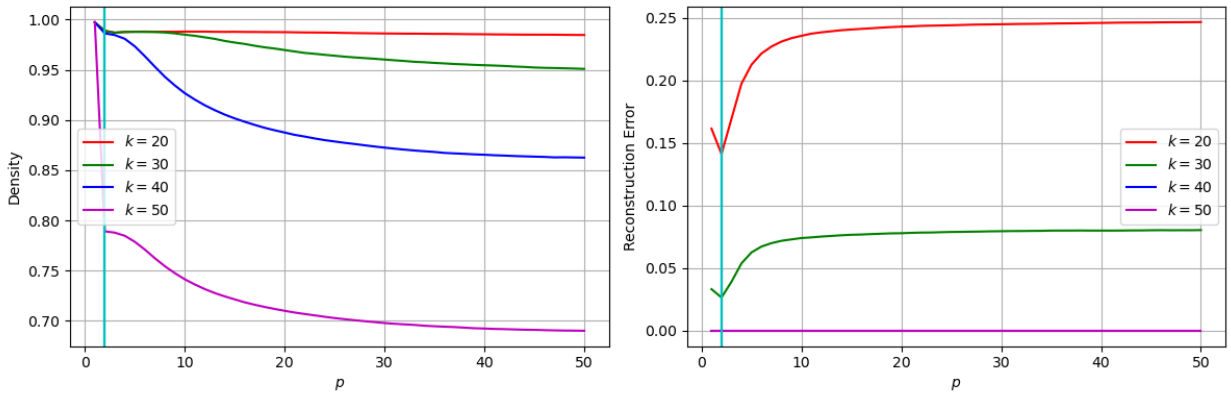

Figure 10: Average sparsity of $\mathbf{Q}^T\mathbf{X}$ and reconstruction error for different values of k on a rank 40 data matrix $\mathbf{X} \in \mathbb{R}^{500\times 50}$. The cyan line indicates $p = 2$ for reference.

Fig. 10 clearly shows a trend that the sparsity increases with p. As expected, $p = 1$ results in the highest density. We note that this is also the case when choosing a value of k. When $k \geq \text{rank}\left(\mathbf{X}\right) = 40$, the reconstruction error is zero for all p tested which suggests that Algorithm 2 finds a $\mathbf{Q}$ whose span contains the subspace of the data for most values of p. When $k < \text{rank}\left(\mathbf{X}\right) = 40$, the sparsity still increases with respect to p, but is overall more

dense than the $k \geq \text{rank}\,(\mathbf{X}) = 40$ case. Furthermore, the reconstruction error also increases with p, which suggests that when setting $p > 2$, there is a trade off between achieving sparsity and returning a $\mathbf{Q}$ that represents the subspace of the data well.

5.2 Clustering Study 1: Synthetic Data

In this section, we test Algorithm 3 on a random data matrix $\mathbf{X} \in \mathbb{R}^{D \times nc}$ that contains c affine subspaces. Each affine subspace corresponds to a class. The random model of the data is

$$\mathbf{X}_i = \mathbf{U}_i \mathbf{Z}_i + 5 \frac{\mathbf{m}_i}{\|\mathbf{m}_i\|_2} \mathbf{1}_n^T \in \mathbb{R}^{D \times n}, \tag{66}$$

$$[\mathbf{Z}_i]_{:,j} \sim \mathcal{N}\left(\mathbf{0}_n, 100\mathbf{I}_n\right), \tag{67}$$

$$\mathbf{m}_i \sim \mathcal{N}\left(\mathbf{0}_D, \mathbf{I}_D\right), \tag{68}$$

$$\mathbf{U}_i \in \mathcal{S}_{D \times n} \text{ s.t. } \mathbf{R}_i \overset{\text{SVD}}{\rightarrow} \mathbf{U}_i \mathbf{\Sigma}_i \mathbf{V}_i^T \text{ and } [\mathbf{R}_i]_{:,j} \sim \mathcal{N}\left(\mathbf{0}_D, \mathbf{I}_D\right), \tag{69}$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \dots & \mathbf{X}_c \end{bmatrix}, \tag{70}$$

for all $i \in \{1, 2, \dots, c\}$ and $j \in \{1, 2, \dots, n\}$. In all of our experimental studies, we use the Fowlkes Mallows Score (FMS) [48]. The FMS is a measure of clustering accuracy that takes values from 0 to 1. A value of 1 indicates perfect clustering and a value of 0 indicates uniformly random labels. The FMS must be used to evaluate clustering performance since the ground truth class labels are generally different from the labels that the k-means algorithm will assign to each cluster. We use the Python implementation of the FMS from [44].

Fig. 11 and 12 show the average FMS for various values of p and D over 1000 independent realizations of $\mathbf{X}$, with $k = 20$, $c = 3$, and $n = 30$. Fig. 11 shows the results of our proposed clustering method in Algorithm 3. Fig. 12 shows the average FMS of spectral clustering via the different methods of forming affinities shown in Table 1, applying k-means directly to the data, and after using L2-PCA to obtain $\mathbf{Q}$ instead of LP1-PCA. For all experiments, we

use the k-means++ [49] method of initialization with a fixed seed of zero.

	Affinity Matrix
$\mathbf{X}$	$\mathbf{W} = \left\lvert \mathbf{X}^T\mathbf{X} \right\rvert$
$\mathbf{Y} = \mathbf{Q}^T\mathbf{X}$	$\mathbf{W} = \left\lvert \mathbf{Y}^T\mathbf{Y} \right\rvert$
$\mathbf{A} = \left\lvert \mathbf{Q}^T\mathbf{X} \right\rvert$	$\mathbf{W} = \mathbf{A}^T\mathbf{A}$

Table 1: Different methods of forming the affinity matrix $\mathbf{W}$ for spectral clustering. The subscript $_{\text{RN}}$ indicates row normalization. The proposed method in Algorithm 3 uses $\mathbf{A}_{\text{RN}}$ to form the affinity matrix.

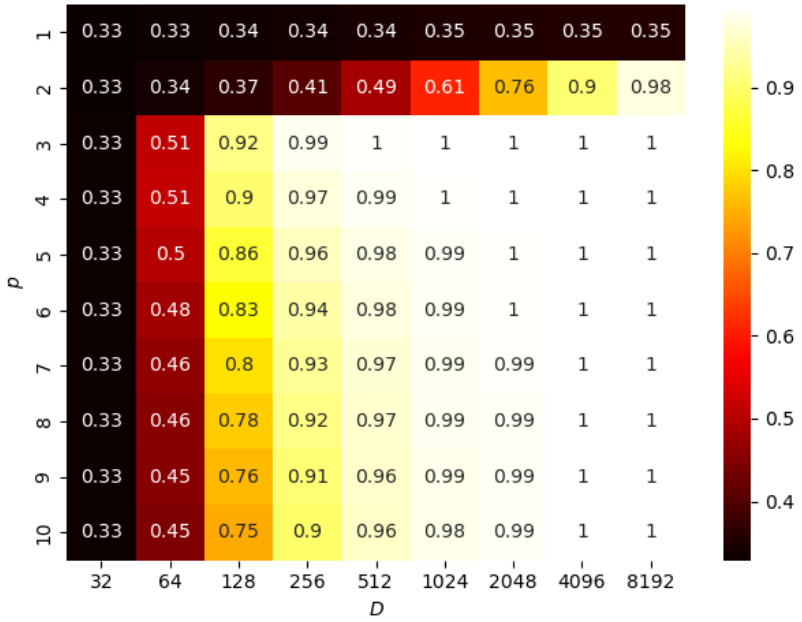

Figure 11: Algorithm 3 average FMS versus D and p on synthetic data in 3 classes with $k = 20$.

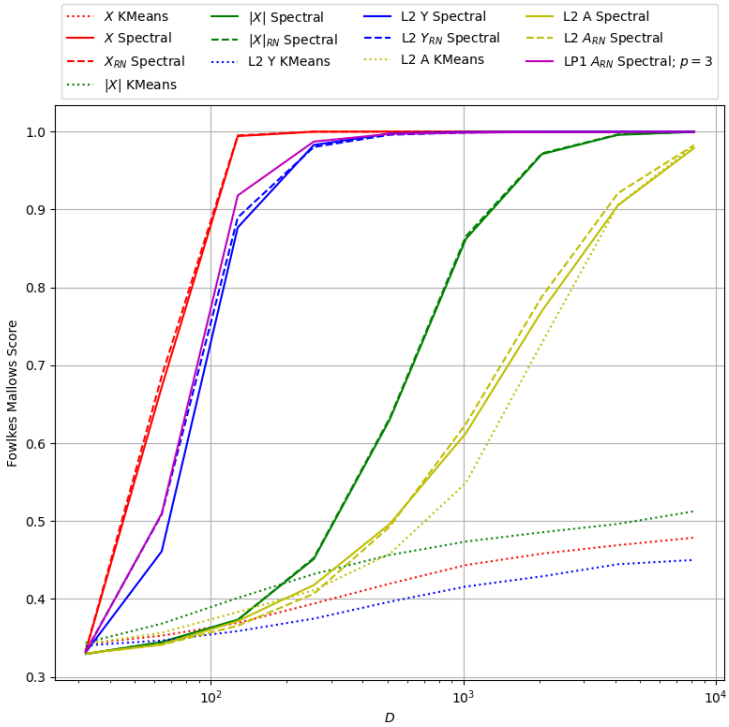

Figure 12: Average FMS score versus D for spectral clustering using affinities from Table 1 and $k = 20$.

The red line in Fig. 12 shows the average FMS achieved by forming the affinity matrix $\left|\mathbf{X}^T\mathbf{X}\right|$ followed by spectral clustering. In this case, D only needs to be 128 in order to achieve nearly perfect clustering. The blue line in Fig. 12 shows the average FMS achieved when L2-PCA is used to find a $\mathbf{Q} \in \mathcal{S}_{D\times k}$ to form the affinity matrix $\left|\mathbf{Y}^T\mathbf{Y}\right|$ where $\mathbf{Y} = \mathbf{Q}^T\mathbf{X}$ for spectral clustering. As D increases, the rank of $\mathbf{X}$ approaches $cn = 90$, so $k = 20$ will typically be much lower than the rank of the data. However, acceptable performance is achieved even when $D = 512$. The trade-off between using a low-rank representation of $\mathbf{X}$ and FMS seems to diminish as D increases asymptotically.

The results of the proposed Algorithm 3 with $k = 20$ in Fig. 11 show that $p = 3$ provides the best FMS over the widest ranges of D while also performing better than the L2-PCA based methods in Fig. 11. Our method achieves perfect clustering when $p = 3$ and $D \geq 512$, and has the advantage of using a low-rank representation of $\mathbf{X}$ when compared to all methods shown in Fig. 12 for $D \geq 512$. The heatmap in Fig. 13 shows one instance of a row normalized $\mathbf{A} = \left|\mathbf{Q}^T\mathbf{X}\right|$ that resulted from Algorithm 3.

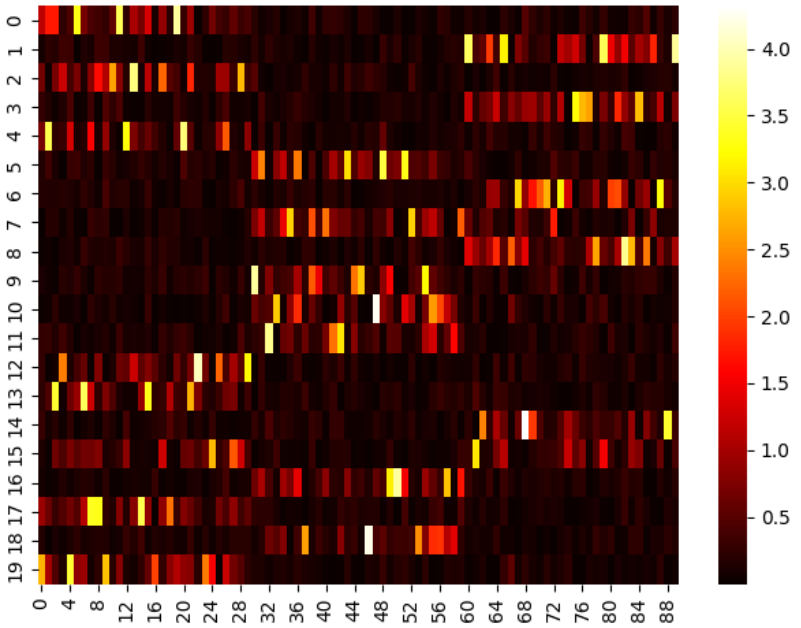

Figure 13: One instance of A_{RN} from Algorithm 3 that displays cluster-ID sparsity.

The heatmap in Fig. 13 displays that the proposed method in Algorithm 3 enforces cluster-ID sparsity. The resulting affinity matrix is shown in Fig. 14 and is block diagonal which achieves a perfect FMS as expected.

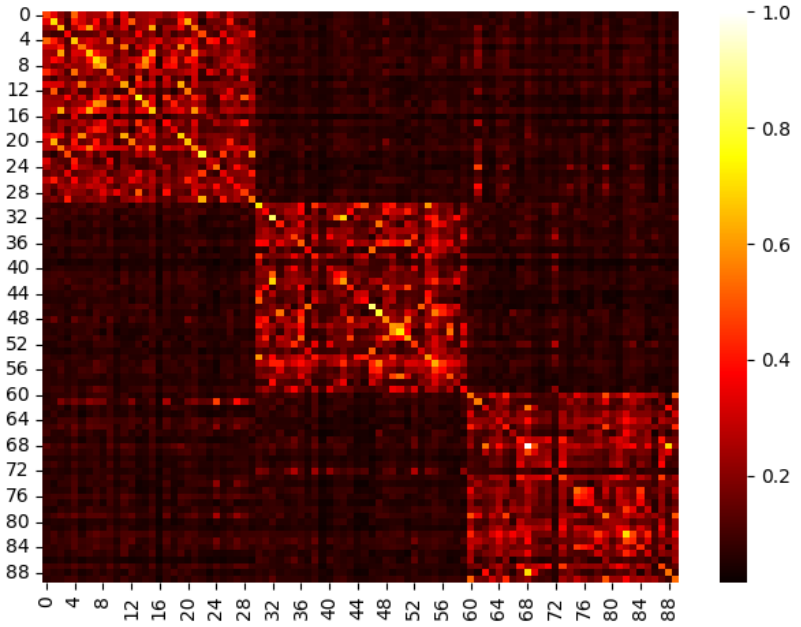

Figure 14: The block diagonal affinity formed from the matrix shown in Fig. 13. This affinity results in perfect clustering via spectral clustering.

5.3 Clustering Study 2: MNIST Dataset

In this study, we attempt to cluster 4 classes from the MNIST dataset [50]. The MNIST database contains 70,000 images of handwritten digits. Each image pixel takes discrete values from 0 to 255. Our data matrix is formed as,

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_0 & \mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_3 \end{bmatrix} \in \mathbb{R}^{784 \times 4N}, \tag{71}$$

where the columns of $\mathbf{X}_c \in \mathbb{R}^{784 \times N}$ contain vectorized images of digit c for all $c \in \{0, 1, 2, 3\}$. Fig. 15 shows the first 4 images of digits 0–3 that were used in this experiment.

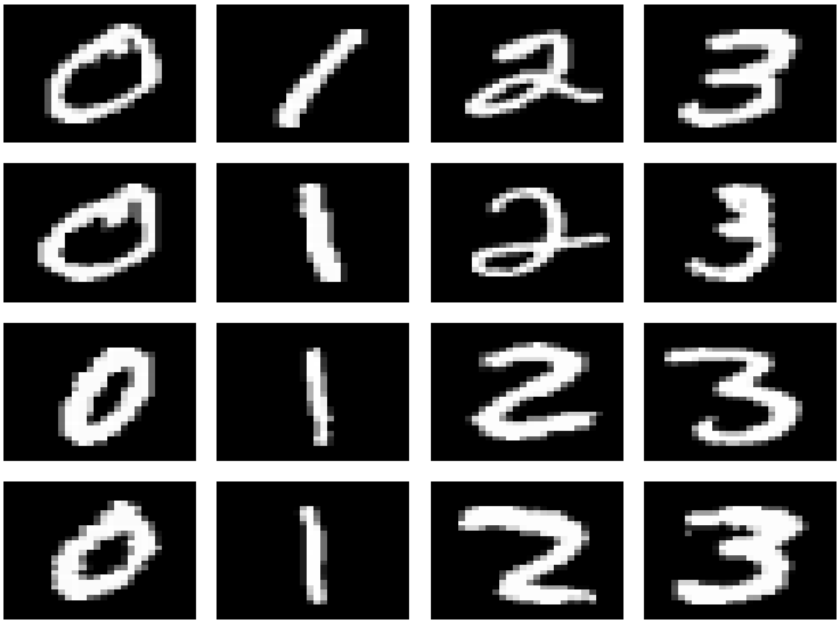

Figure 15: MNIST dataset digits 0–3.

In this study, we have arbitrarily chosen the first $N = 30$ samples of each digit. We have chosen a smaller N since our method requires that the rank of the data must be small compared to the ambient dimension. Next, consider the singular value decomposition of each $\mathbf{X}_c$, i.e., $\mathbf{U}_c\mathbf{\Sigma}_c\mathbf{V}_c \overset{\text{SVD}}{\leftarrow} \mathbf{X}_c$ for $c \in \{0, 1, 2, 3\}$. We also define the singular value decomposition of the full data matrix $\mathbf{X}$ as $\mathbf{U}\mathbf{\Sigma}\mathbf{V} \overset{\text{SVD}}{\leftarrow} \mathbf{X}$. Fig. 16 shows a plot of the nuclear norm versus rank for each $\mathbf{X}_c$. The nuclear norm is defined as $\sum_{i=1}^{r} \sigma_i$, where r is the rank.

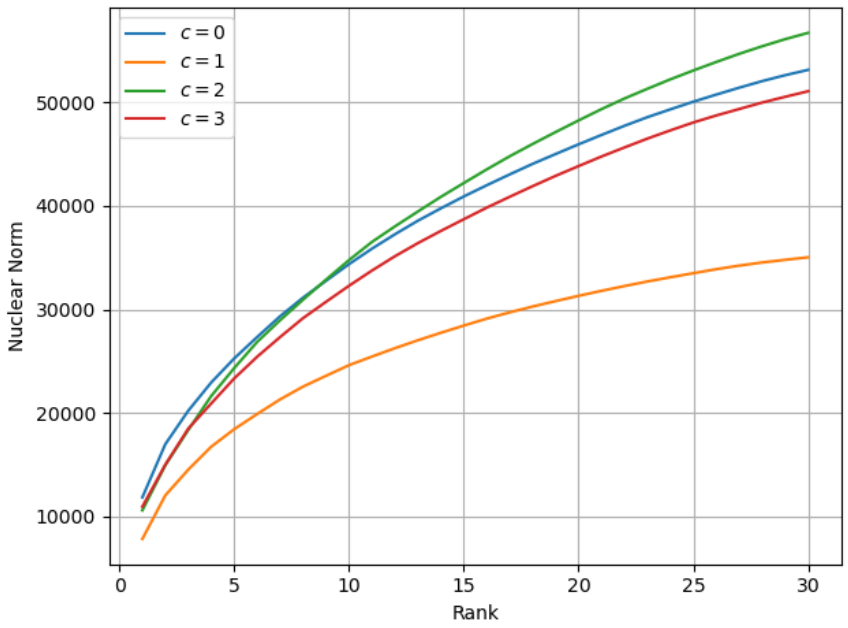

Figure 16: Nuclear norm of each matrix $\mathbf{X}_c$ versus rank for MNIST.

The nuclear norm curves in Fig. 16 do not plateau, which indicates that each $\mathbf{X}_c$ is full rank, i.e., $\text{rank}\,(\mathbf{X}_c) = 30$ for $c \in \{0, 1, 2, 3\}$. We can also compare the subspaces of each class. The heatmap in Fig. 17 shows the normalized subspace distance defined in (34) between $\mathbf{U}_i$ and $\mathbf{U}_j$ for $(i, j) \in \{0, 1, 2, 3\}^2$.

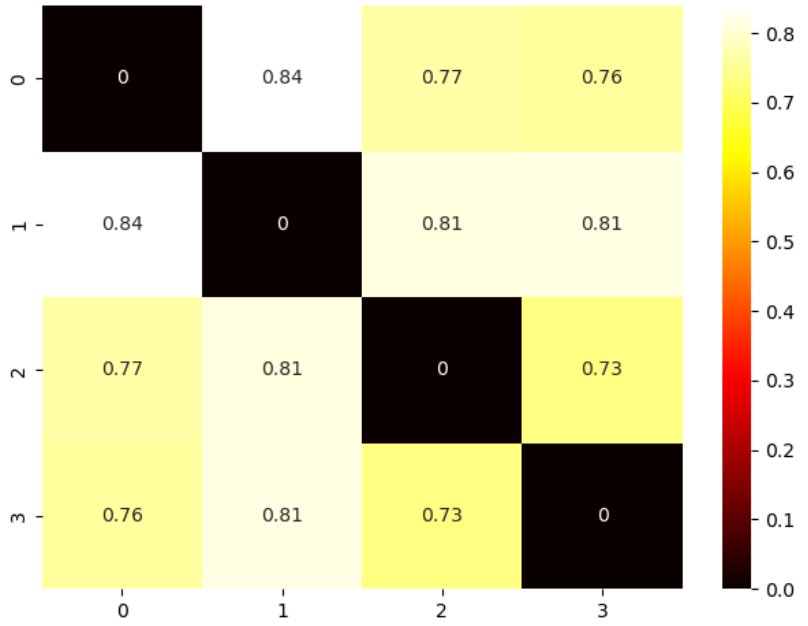

Figure 17: Normalized subspace distance between classes in MNIST. The (i, j)-th entry of the heatmap shows the normalized subspace distance defined in (34) between class i and j.

Fig. 17 shows that the subspaces formed by different digits have a small amount of overlap. Although this overlap is small, it could still have a significant affect on clustering performance if the intersection of such subspaces contributes heavily to the norms of each $\mathbf{x}_i$. With this in mind, we use the affinity matrix $\mathbf{W} = \left|\mathbf{Y}^T\mathbf{Y}\right|$ with $\mathbf{Y} = \mathbf{U}_{:,m:n}^T\mathbf{X}$ as input to spectral clustering for $(m, n) \in \{1, \ldots, cN\}$ and $m \leq n$. The value of m controls how many of the top singular vectors are to be discarded and the value of $n - m$ is the number of subsequent singular vectors that are kept. The results of such experiment is shown in Fig. 18 where the heatmap shows the FMS. Fig. 18 suggests that the first two singular vectors of $\mathbf{U}$ should be discarded to achieve better clustering performance. Conversely, the third singular vector of $\mathbf{U}$ seems to be essential for good performance.

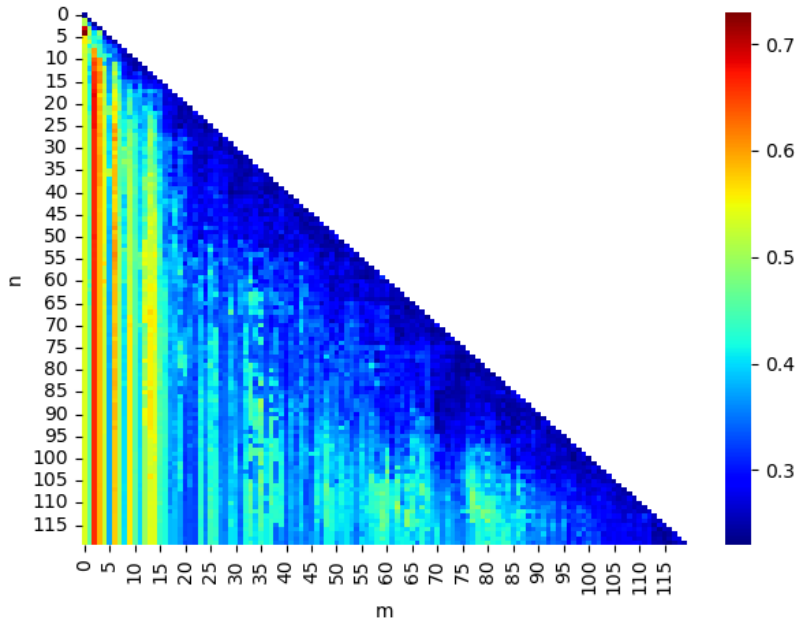

Figure 18: How subspace intersection in MNIST affects clustering performance. The (n, m)-th entry of the heatmap shows the FMS that results from spectral clustering with the affinity matrix $\mathbf{W} = \left|\mathbf{Y}^T\mathbf{Y}\right|$ and $\mathbf{Y} = \mathbf{U}_{:,m:n}^T\mathbf{X}$.

However, in practice, such preprocessing must be done using trial and error which might not be viable.

Fig. 19 shows a comparison of the FMS resulting from several algorithms applied to $\mathbf{X}$ with $k = 10$ and $p = 3$. Several methods of forming the affinity matrix as shown in Table 1 were used in addition to using the $\mathbf{Q}^T\mathbf{X}$ from both L2-PCA and LP1-PCA. We have also compared our results with k-means applied directly to the data and the popular subspace clustering algorithms LSA [11] and SSC [9].

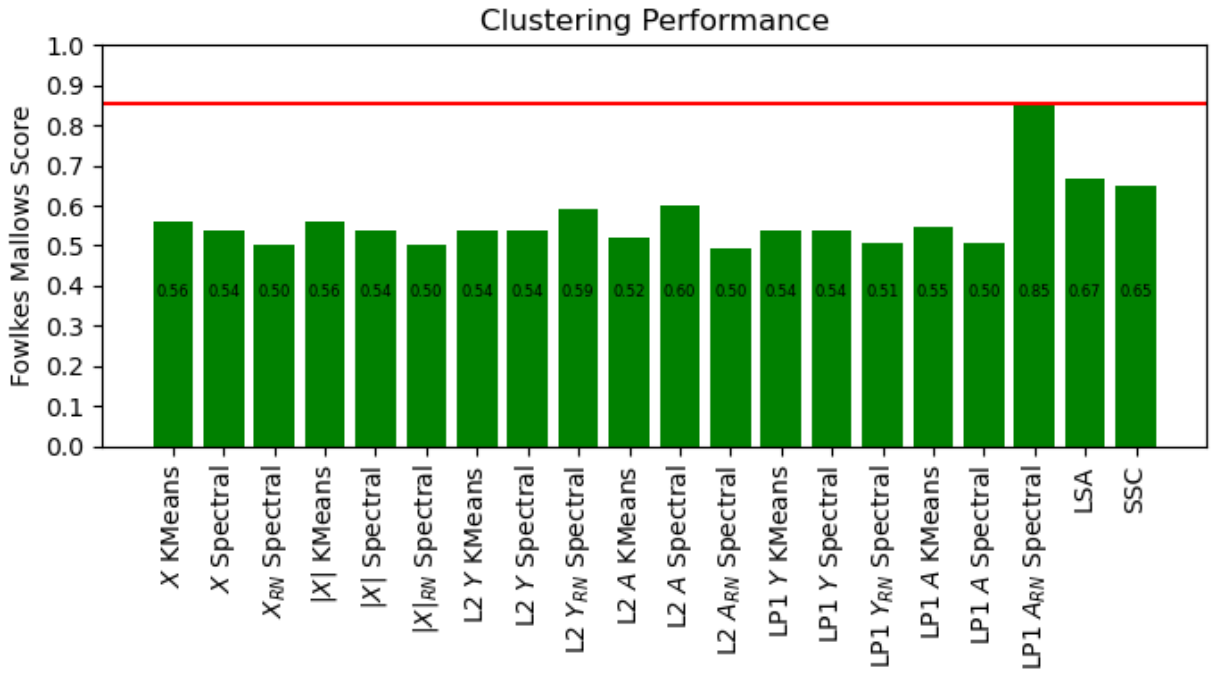

Figure 19: FMS of various clustering methods on MNIST which include: spectral clustering with different affinities as described in Table 1, k-means applied directly to the data, SSC, and LSA. The parameter $k = 10$ was used for all PCA methods and $p = 3$ was used for LP1-PCA.

Fig. 19 shows that the proposed method in Algorithm 3 has superior clustering performance with this particular dataset. The row normalized $\mathbf{A} = \mathbf{Q}^T\mathbf{X}$ resulting from Algorithm 3 is shown in Fig. 20 which mostly exhibits cluster-ID sparsity with the exception of the first row. The first row likely is the cause of the cluster errors.

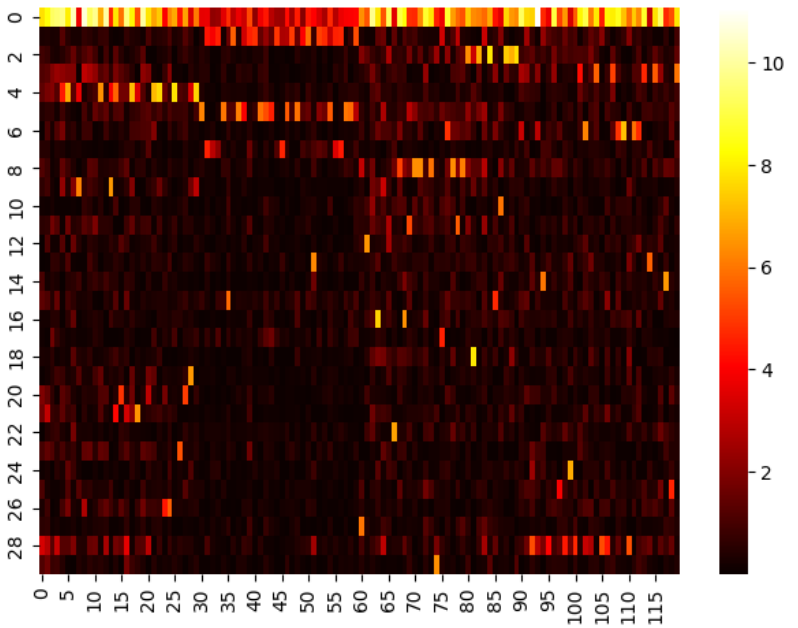

Figure 20: The row normalized $\mathbf{A} = \mathbf{Q}^T\mathbf{X}$ resulting from Algorithm 3 in Fig. 19.

The affinity matrix formed from Fig. 20 is shown in Fig. 21. This affinity matrix appears to be able to separate digit 0 and 1, but it has off diagonal blocks associated with the other digits that cause clustering errors.

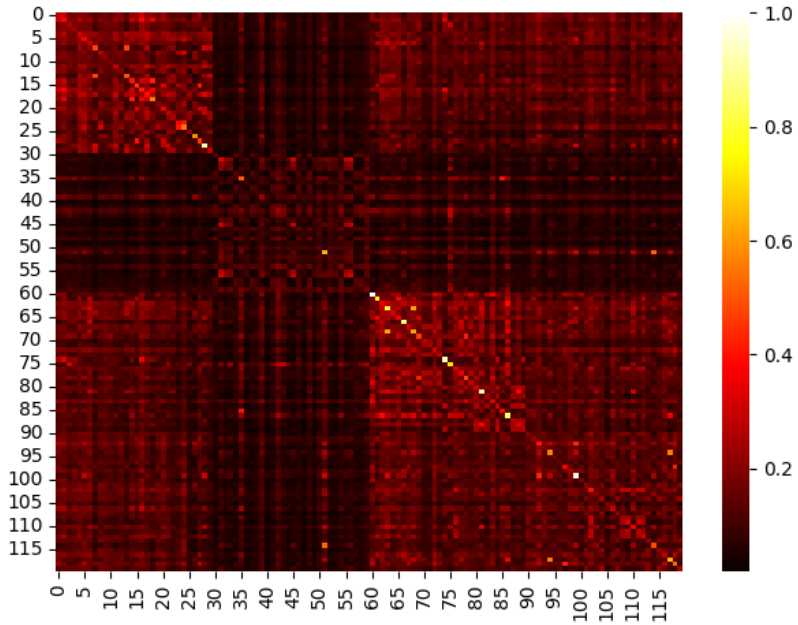

Figure 21: The resulting affinity formed from Fig. 20.

5.4 Clustering Study 3: Cropped Extended Yale Face B Dataset

For this experiment, we attempt to cluster 4 classes from the Cropped Extended Yale Face B (YALE) dataset [51]. The YALE dataset contains images of 38 human subjects under 64 illumination conditions that are cropped to have a size of 168 by 192. Each pixel takes a discrete value from 0 to 255. Our data matrix is formed as

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_0 & \mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_3 \end{bmatrix} \in \mathbb{R}^{32256 \times 4N}, \tag{72}$$

where the columns of $\mathbf{X}_c \in \mathbb{R}^{32256 \times N}$ are the vectorized images of class c for all $c \in \{0, 1, 2, 3\}$. In this study, we have arbitrarily chosen subjects 4, 5, 6, and 7 for our classes, and chosen all $N = 64$ lighting conditions for each subject as samples. Fig. 22 shows the first 4 images of each the subjects used in this experiment.

Figure 22: YALE dataset subjects 4–7.

The number of samples N was chosen to be much lower than the ambient dimension. Once again, consider the singular value decomposition of each $\mathbf{X}_c$, i.e., $\mathbf{U}_c\mathbf{\Sigma}_c\mathbf{V}_c \overset{\text{SVD}}{\leftarrow} \mathbf{X}_c$ for $c \in \{0, 1, 2, 3\}$. Similarly, we define the singular value decomposition of the full data matrix

$\mathbf{X}$ as $\mathbf{U}\mathbf{\Sigma}\mathbf{V} \overset{\text{SVD}}{\leftarrow} \mathbf{X}$. Fig. 23 shows a plot of the nuclear norm versus rank for each $\mathbf{X}_c$.

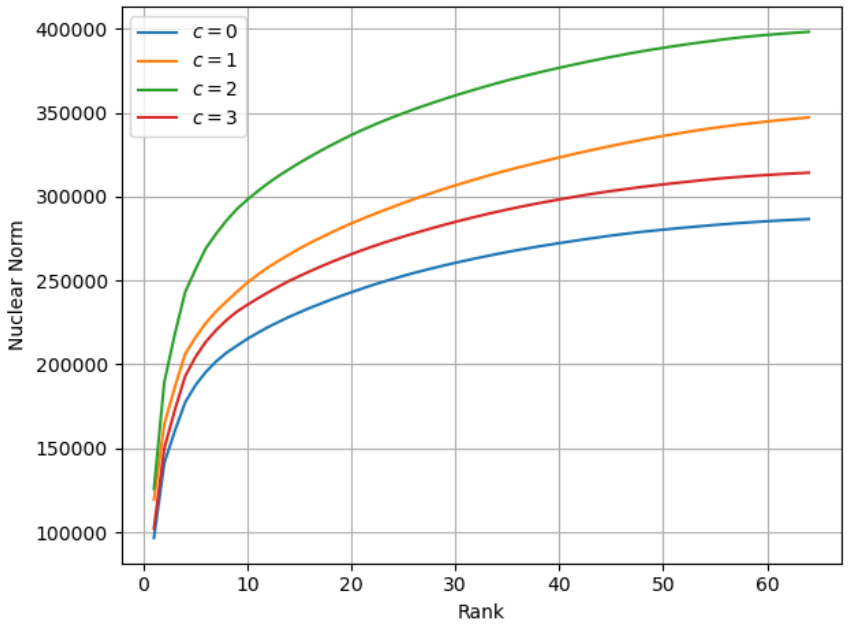

Figure 23: Nuclear norm of each matrix $\mathbf{X}_c$ versus rank for YALE.

The nuclear norm curves in 23 show that the class data matrices $\mathbf{X}_c$ are full rank, i.e., $\text{rank}\,(\mathbf{X}_c) = 30$ for $c \in \{0, 1, 2, 3\}$. This is a result of $N << D$. Again, we can also compare the subspaces of each class. The heatmap in Fig. 24 shows the normalized subspace distance defined in (34) between $\mathbf{U}_i$ and $\mathbf{U}_j$ for $(i, j) \in \{0, 1, 2, 3\}^2$.

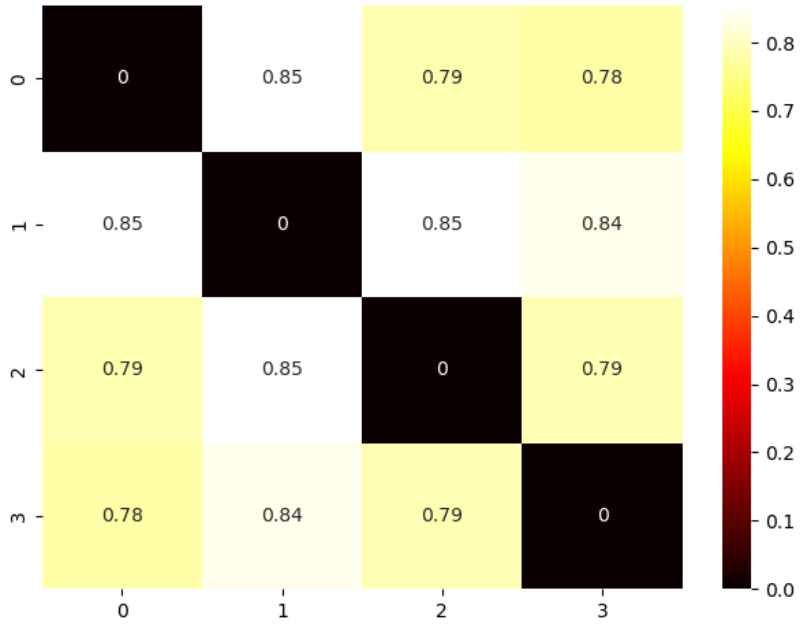

Figure 24: Normalized subspace distance between classes in YALE. The (i, j)-th entry of the heatmap shows the normalized subspace distance defined in (34) between class i and j.

Fig. 24 shows that there is some degree of intersection between the different class subspaces. Again, we investigate how this subspace overlap affects clustering performance. The affinity matrix $\mathbf{W} = \left|\mathbf{Y}^T\mathbf{Y}\right|$ with $\mathbf{Y} = \mathbf{U}_{:,m:n}^T\mathbf{X}$ is again used as input to spectral clustering for $(m, n) \in \{1, \ldots, cN\}$ and $m \leq n$. The value of m controls how many of the top singular vectors are to be discarded and the value of $n - m$ is the number of subsequent singular vectors that are kept. Fig. 25 shows a heatmap of the resulting FMS scores.

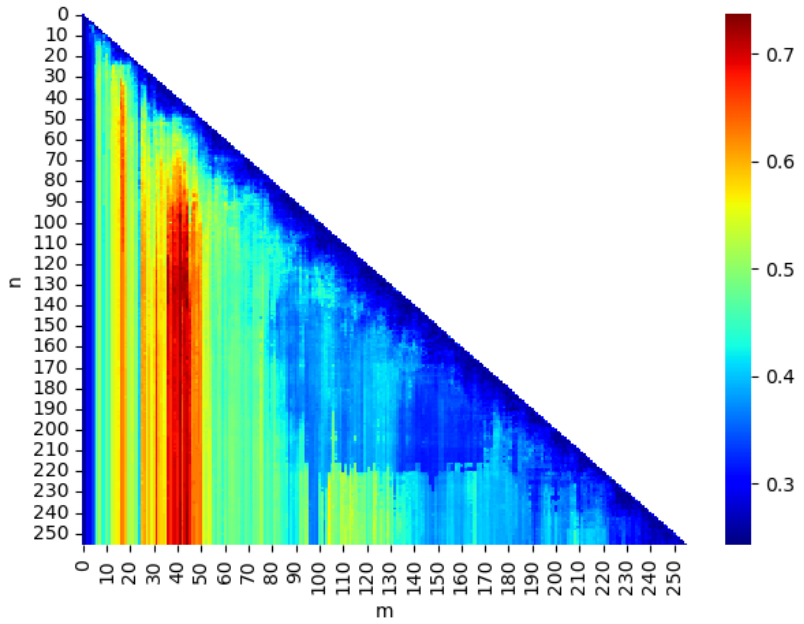

Figure 25: How subspace intersection in YALE affects clustering performance. The (n, m)-th entry of the heatmap shows the FMS that results from spectral clustering with the affinity matrix $\mathbf{W} = |\mathbf{Y}^T\mathbf{Y}|$ and $\mathbf{Y} = \mathbf{U}_{:,m:n}^T\mathbf{X}$.

Fig. 25 indicates that the first 5 singular vectors of $\mathbf{U}$ make it nearly impossible to perform clustering. The red regions in Fig. 25 indicate regions of good clustering performance. For regions of m from 35 to 50, a wide range of n will yield good performance. There is another small region around $m = 20$ that gives good performance for a wide range of n. Overall, this suggests our classes share a subspace that contributes significantly to the norms of each point which hinders clustering performance.

For the next experiment on YALE, we define the new data matrix

$$\tilde{\mathbf{X}} = \mathbf{U}_{:,20:}\mathbf{U}_{:,20:}^T\mathbf{X}, \tag{73}$$

where $\tilde{\mathbf{X}}$ is our original data $\mathbf{X}$ that has been projected onto the null-space of the first 19 singular vectors of $\mathbf{U}$. Our previous experiments without the null-space projection resulted in poor performance for most of the methods that we tried. Fig. 26 shows the FMS of several different clustering methods applied to $\tilde{\mathbf{X}}$ with $k = 110$ and $p = 16$, which include: spectral clustering with different affinities as described in Table 1 and k-means applied directly to

the data.

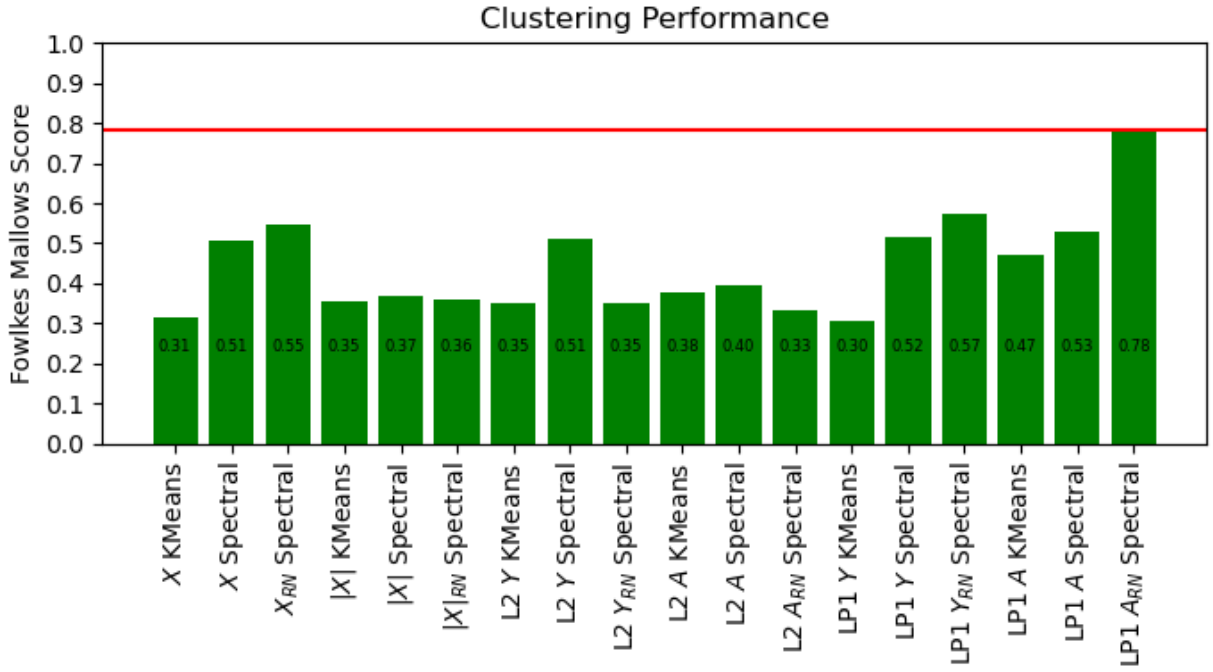

Figure 26: FMS of various clustering methods on YALE which include: spectral clustering with different affinities as described in Table 1, k-means applied directly to the data. The parameter $k = 110$ was used for all PCA methods and $p = 16$ was used for LP1-PCA.

The results in Fig. 26 indicate that the proposed method in Algorithm 3 is able to outperform the other methods tested for this particular data matrix. Fig. 27 shows the row normalized matrix $\mathbf{A} = \left|\mathbf{Q}^T\mathbf{X}\right|$ with Fig. 28 showing the corresponding affinity matrix.

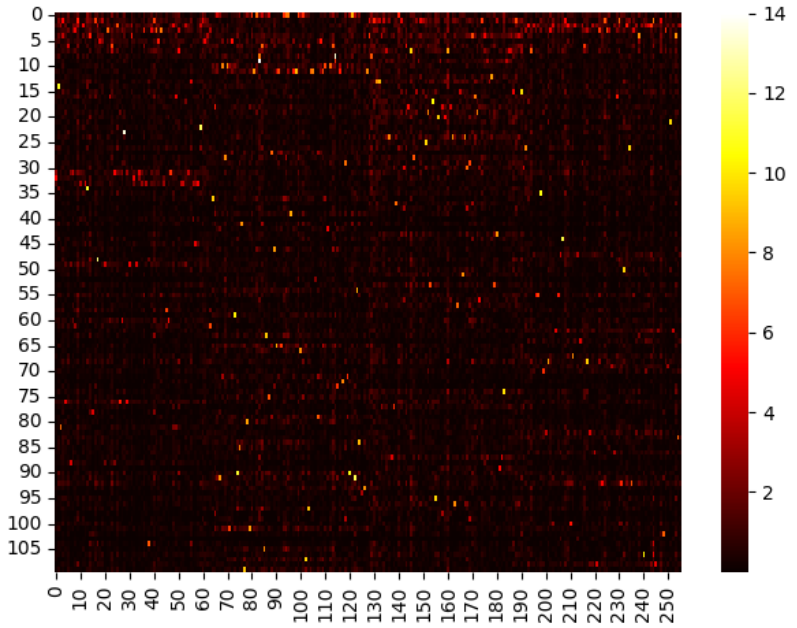

Figure 27: The row normalized $\mathbf{A} = \mathbf{Q}^T\mathbf{X}$ resulting from Algorithm 3 in Fig. 26.

The matrix in Fig. 27 is extremely sparse. It has cluster-ID sparsity, but also has the problem that columns of the same class look orthogonal. This results in a disconnected affinity matrix as shown in Fig. 28. That being said, the affinity in Fig. 28 does show some degree of block diagonal structure. This could explain why the FMS is still very good compared to competing methods in this case.

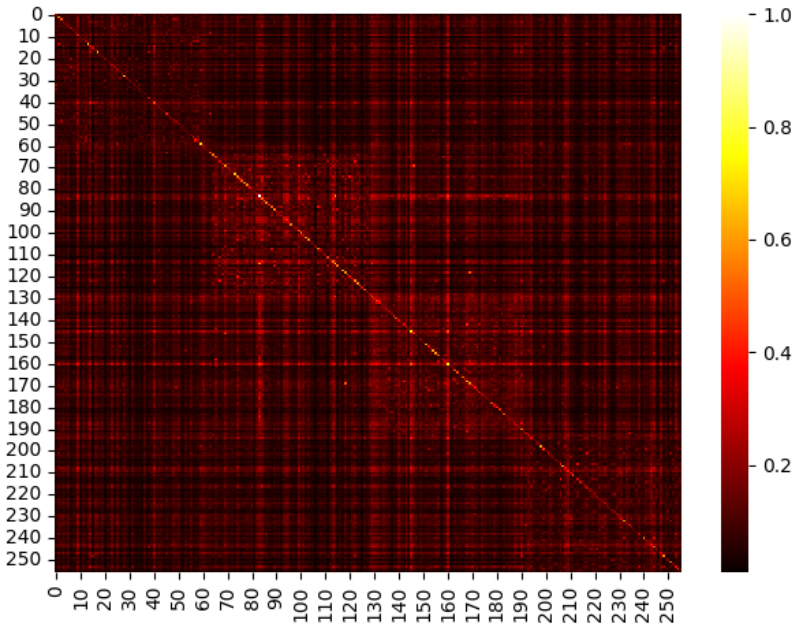

Figure 28: The resulting affinity formed from Fig. 27.

6 Conclusion

As big data continues to evolve, so does the demand for being able to extract information from high-dimensional data. In this work, we have shown that for the task of unsupervised subspace clustering, high-dimensionality can be an advantage given that the data is of low-rank. To support this claim, we have identified a statistical model of data that can achieve perfect clustering when the ambient dimension is high compared to the rank of the clusters, and introduced the concept of cluster-ID sparsity. In order to enforce cluster-ID sparsity on real datasets, we proposed the method of LP1-PCA Spectral Clustering, which provides a simple method for performing subspace clustering when compared to other methods such as SSC, LSA, and SCC. Furthermore, we have shown some examples of datasets formed from YALE and MNIST where LP1-PCA can outperform such methods.

In addition, we have developed a novel iterative algorithm to approximately solve LP1-PCA for general $p \geq 1$. To the best of our knowledge, no such algorithm exists in the literature at the time of writing. However, it is worth mentioning that our algorithm corre-

sponds to the algorithm of Nie et al. [30] when $p = 1$ and [33] when $p = 2$. While the general LP1-PCA problem is NP-hard, we also provide an exact solution for a rank one data matrix for general k and p. While the rank one solution is not practical for a real-world data, it gives us insight into how the selection of p changes the solution of LP1-PCA, namely, that $p \in [1, 2)$ induces density and $p \in (2, \infty)$ induces sparsity in the projection components. We show extensive empirical evidence to support the later claim.

While our work has shown promise in the areas of subspace clustering, many unanswered questions remain that deserve more research effort. One of such questions regards to how one selects the value of p. While we show that $p > 2$ helps create cluster-ID sparsity, we still do not fully understand yet how changing p on this interval affects the clustering performance. In a similar manner, we were able to show that one needs to set k to be less than the rank of the data to take advantage of cluster-ID sparsity, but we do not yet know how k affects the clustering performance more generally. Lastly, we lack the theoretical proofs and guarantees that show how processing the matrix $\mathbf{Q}^T\mathbf{X}$ via row or column normalization affects the general clustering performance. One could also develop different methods of pruning $\mathbf{Q}^T\mathbf{X}$. In this thesis, we have laid solid foundations of a very promising new research area which we believe that deserves to be further investigated.

References

[1] T. E. Boult and L. G. Brown, "Factorization-based segmentation of motions," in *Proceedings of the IEEE workshop on visual motion*, pp. 179–180, IEEE Computer Society, 1991.

[2] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *International Journal of Computer Vision*, vol. 29, no. 3, pp. 159–179, 1998.

[3] C. W. Gear, "Multibody grouping from motion images," *International Journal of Computer Vision*, vol. 29, no. 2, pp. 133–150, 1998.

[4] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (gpca)," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 12, pp. 1945–1959, 2005.

[5] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[6] Y. Ma, H. Derksen, W. Hong, and J. Wright, "Segmentation of multivariate mixed data via lossy data coding and compression," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 9, pp. 1546–1562, 2007.

[7] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[8] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 14, 2001.

[9] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[10] G. Chen and G. Lerman, "Spectral curvature clustering (scc)," *International Journal of Computer Vision*, vol. 81, no. 3, pp. 317–330, 2009.

[11] J. Yan and M. Pollefeys, "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate," in *European conference on computer vision*, pp. 94–106, Springer, 2006.

[12] A. Goh and R. Vidal, "Segmenting motions of different types by unsupervised manifold clustering," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–6, IEEE, 2007.

[13] G. Liu, Z. Lin, Y. Yu, *et al.*, "Robust subspace segmentation by low-rank representation.," in *Icml*, vol. 1, p. 8, Citeseer, 2010.

[14] R. Vidal, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011.

[15] E. M. Stein and R. Shakarchi, *Princeton lectures in analysis*. Princeton University Press Princeton, 2011.

[16] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[17] S. Chen and D. Donoho, "Basis pursuit," in *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 41–44, IEEE, 1994.

[18] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[19] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, vol. 2, no. 11, pp. 559–572, 1901.

[20] C. Ding and X. He, "K-means clustering via principal component analysis," in *Proceedings of the twenty-first international conference on Machine learning*, p. 29, 2004.

[21] G. Shaw and D. Manolakis, "Signal processing for hyperspectral image exploitation," *IEEE Signal processing magazine*, vol. 19, no. 1, pp. 12–16, 2002.

[22] Q. Du and J. E. Fowler, "Hyperspectral image compression using jpeg2000 and principal component analysis," *IEEE Geoscience and Remote sensing letters*, vol. 4, no. 2, pp. 201–205, 2007.

[23] H. Hotelling, "Analysis of a complex of statistical variables into principal components.," *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[24] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[25] N. Kwak, "Principal component analysis based on l1-norm maximization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 9, pp. 1672–1680, 2008.

[26] P. P. Markopoulos, S. Kundu, S. Chamadia, and D. A. Pados, "Efficient l1-norm principal-component analysis via bit flipping," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4252–4264, 2017.

[27] P. P. Markopoulos, G. N. Karystinos, and D. A. Pados, "Optimal algorithms for l1-subspace signal processing," *IEEE Transactions on Signal Processing*, vol. 62, no. 19, pp. 5046–5058, 2014.

[28] P. P. Markopoulos, M. Dhanaraj, and A. Savakis, "Adaptive l1-norm principal-component analysis with online outlier rejection," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 6, pp. 1131–1143, 2018.

[29] J. P. Brooks, J. H. Dulá, and E. L. Boone, "A pure l1-norm principal component analysis," *Computational statistics & data analysis*, vol. 61, pp. 83–98, 2013.

[30] F. Nie, H. Huang, C. Ding, D. Luo, and H. Wang, "Robust principal component analysis with non-greedy l1-norm maximization," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[31] B. Minnehan and A. Savakis, "Grassmann manifold optimization for fast l1-norm principal component analysis," *IEEE Signal Processing Letters*, vol. 26, no. 2, pp. 242–246, 2018.

[32] M. Johnson and A. Savakis, "L1-grassmann manifolds for robust face recognition," in *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 482–486, IEEE, 2015.

[33] F. Nie and H. Huang, "Non-greedy l21-norm maximization for principal component analysis," *arXiv preprint arXiv:1603.08293*, 2016.

[34] C. Ding, D. Zhou, X. He, and H. Zha, "R 1-pca: rotational invariant l 1-norm principal component analysis for robust subspace factorization," in *Proceedings of the 23rd international conference on Machine learning*, pp. 281–288, 2006.

[35] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 1–37, 2011.

[36] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *2003 IEEE Computer Society Con-*

ference on Computer Vision and Pattern Recognition, 2003. Proceedings., vol. 1, pp. I–I, IEEE, 2003.

[37] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 212–225, 2008.

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

[39] K.-i. Kanatani, "Motion segmentation by subspace separation and model selection," in *Proceedings Eighth IEEE International Conference on computer Vision. ICCV 2001*, vol. 2, pp. 586–591, IEEE, 2001.

[40] Y. Wu, Z. Zhang, T. S. Huang, and J. Y. Lin, "Multibody grouping via orthogonal subspace decomposition," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 2, pp. II–II, IEEE, 2001.

[41] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[42] F. R. Chung and F. C. Graham, *Spectral graph theory*. No. 92, American Mathematical Soc., 1997.

[43] P. E. Hart, D. G. Stork, and R. O. Duda, *Pattern classification*. Wiley Hoboken, 2000.

[44] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[45] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[46] M. R. Garey and D. S. Johnson, *Computers and intractability*, vol. 174. freeman San Francisco, 1979.

[47] P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.

[48] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *Journal of the American statistical association*, vol. 78, no. 383, pp. 553–569, 1983.

[49] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," tech. rep., Stanford, 2006.

[50] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

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