SIMILARITY MODELING FOR MACHINE LEARNING

BY

YINGZHEN YANG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Doctoral Committee:

Professor Thomas S. Huang, Chair
Professor Mark Hasegawa-Johnson
Professor Zhi-Pei Liang
Dr. Jianchao Yang, Snap Inc.

ABSTRACT

Similarity is the extent to which two objects resemble each other. Modeling similarity is an important topic for both machine learning and computer vision. In this dissertation, we first propose a discriminative similarity learning method, then introduce two novel sparse similarity modeling methods for high dimensional data from the perspective of manifold learning and subspace learning. Our sparse similarity modeling methods learn sparse similarity and consequently generate a sparse graph over the data. The generated sparse graph leads to superior performance in clustering and semi-supervised learning, compared to existing sparse graph based methods such as ℓ^1-graph and Sparse Subspace Clustering (SSC).

More concretely, our discriminative similarity learning method adopts a novel pairwise clustering framework by bridging the gap between clustering and multiclass classification. This pairwise clustering framework learns an unsupervised nonparametric classifier from each data partition, and searches for the optimal partition of the data by minimizing the generalization error of the learned classifiers associated with the data partitions.

Regarding to our sparse similarity modeling methods, we propose a novel ℓ^0 regularized ℓ^1-graph (ℓ^0-ℓ^1-graph) to improve ℓ^1-graph from the perspective of manifold learning. Our ℓ^0-ℓ^1-graph generates a sparse graph that is aligned to the manifold structure of the data for better clustering performance. From the perspective of learning the subspace structures of the high dimensional data, we propose ℓ^0-graph that generates a subspace-consistent sparse graph for clustering and semi-supervised learning. Subspace-consistent sparse graph is a sparse graph where a data point is only connected to other data that lie in the same subspace, and the representative method Sparse Subspace Clustering (SSC) proves to generate subspace-consistent sparse graph under certain assumptions on the subspaces and the data, e.g. independent/disjoint subspaces and subspace incoherence/affinity. In contrast, our ℓ^0-graph can generate subspace-consistent sparse graph for arbitrary distinct underlying subspaces under far less restrictive assump-

tions, i.e. only i.i.d. random data generation according to arbitrary continuous distribution. Extensive experimental results on various data sets demonstrate the superiority of ℓ^0-graph compared to other methods including SSC for both clustering and semi-supervised learning.

The proposed sparse similarity modeling methods require sparse coding using the entire data as the dictionary, which can be inefficient especially in case of large-scale data. In order to overcome this challenge, we propose Support Regularized Sparse Coding (SRSC) where a compact dictionary is learned. The data similarity induced by the support regularized sparse codes leads to compelling clustering performance. Moreover, a feed-forward neural network, termed Deep-SRSC, is designed as a fast encoder to approximate the codes generated by SRSC, further improving the efficiency of SRSC.

To my parents, for their love and support.

ACKNOWLEDGMENTS

First, I would like to express my deep gratitude to my adviser Professor Thomas S. Huang for all of his guidance, advice, and support throughout my Ph.D. study at UIUC. It would be very difficult for me to devote most of my work time to my favorite and fundamental topics of statistical machine learning research without his enlightened attitude and support. During the past five and half years, I have been inspired by his vision, wisdom, passion and dedication to high-quality research and his respectful personality. The experience of working with him is undoubtedly an invaluable merit for my professional career in the future.

It has been a great honor for me to collaborate with Professor Feng Liang at the Department of Statistics of UIUC, Shuicheng Yan at National University of Singapore. I was deeply impressed by their profound insights into my research problems. I was lucky to work with Dr. Nebojsa Jojic as a summer intern at Microsoft Research Redmond. From him, I learned research skills and how to view my research topics from different perspectives. I am very thankful to Dr. Pushmeet Kohli at Microsoft Research Redmond for his suggestions on my research. I also appreciate the comments and advice from Professor Mark Hasegawa-Johnson and Professor Zhi-Pei Liang, and I am fortunate to have them as my thesis committee members.

I would like to extend my appreciation to my mentor, Mr. Dan Gelb, for his suggestions and advice on my research project at Hewlett-Packard Laboratories during the summer of 2011. I also learned a lot of engineering skills from him. I greatly thank Dr. Jianchao Yang at Snapchat Research for his insightful comments and advice on my various research projects. I owe sincere gratitude to Professor Jiashi Feng at National University of Singapore. I also owe sincere thanks to many student collaborators including IFP members Wei Han, Zhaowen Wang, Jiangping Wang and Jiahui Yu, as well as Zhiding Yu from Carnegie Mellon University.

I would also like to extend my deepest gratitude to my family. Without their encouragement and unconditional love, life would not be so beautiful.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Similarity is the extent to which two objects resemble each other. Similarity modeling is regarded as one of the most important topics in machine learning with broad applications in computer vision exhibiting compelling performance in various learning and vision tasks. In this dissertation, we first propose a discriminative similarity learning method, then introduce two novel similarity modeling methods for general high dimensional data from the perspective of manifold learning and subspace learning with convincing theoretical and empirical results. We finally propose Support Regularized Sparse Coding, which learns a compact dictionary rather than using the entire data as the dictionary in the previous two methods.

A discriminative similarity learning method is proposed in Chapter 2, which adopts a new framework for pairwise clustering wherein the pairwise similarity is derived as the generalization error bound for the unsupervised nonparametric classifier. The unsupervised classifier is learned from unlabeled data and the hypothetical labeling. The quality of the hypothetical labeling is measured by the associated generalization error of the learned classifier, and the hypothetical labeling with minimum associated generalization error bound is preferred. We consider two nonparametric classifiers, i.e. the nearest neighbor classifier (NN) and the plug-in classifier (or the kernel density classifier). The generalization error bounds for both unsupervised classifiers are expressed as sum of pairwise terms between the data points, which can be interpreted as nonparametric pairwise similarity measure between the data points. Under uniform distribution, both nonparametric similarity measures exhibit a well known form of kernel similarity. We also prove that the generalization error bound for the unsupervised plug-in classifier is asymptotically equal to the weighted volume of cluster boundary [1] for Low Density Separation, a widely used criterion for semi-supervised learning and clustering.

Sparse representation serves as the major tool for our sparse similarity modeling methods. Chapter 3 and 4 describe our similarity learning models that learn sparse

similarity and consequently generate a sparse graph over which superior clustering and semi-supervised learning performance are achieved. A sparse graph is a graph which has only a few edges of nonzero weights for each vertex, wherein the learned sparse similarity serves as the edge weight. Sparse graph is demonstrated to be effective for clustering and semi-supervised learning, especially for high dimensional data. Examples of sparse graph based machine learning methods include ℓ^1-graph [2, 3] and Sparse Subspace Clustering (SSC) [4]. Our similarity learning models produce an improved sparse graph from the perspective of manifold learning and subspace learning.

More concretely, Chapter 3 describes similarity learning from the perspective of manifold learning. We propose a novel ℓ^0 regularized ℓ^1-graph (ℓ^0-ℓ^1-graph) to improve ℓ^1-graph. Our ℓ^0-ℓ^1-graph generates a sparse graph that is aligned to the manifold structure of the data for better clustering performance based on manifold assumption [5, 6, 7, 8, 9]. ℓ^0-ℓ^1-graph employs manifold assumption on the local structure of the sparse graph, which requires that nearby data in the manifold are encouraged to have similar local sparse graph structure, i.e. they should have similar neighbors and similar edge weights in the sparse graph. ℓ^0-ℓ^1-graph imposes such manifold assumption by using ℓ^0-distance between the sparse codes in the graph regularization term. We develop an iterative proximal method to solve the nonconvex optimization problem of ℓ^0-ℓ^1-graph with proven guarantee of convergence. Extensive experimental results on various real data sets demonstrate the superiority of ℓ^0-ℓ^1-graph over other competing clustering methods including ℓ^1-graph and its ℓ^2 regularized version, namely ℓ^2-ℓ^1-graph. ℓ^2-ℓ^1-graph uses ℓ^2-distance for the sparse codes in the graph regularization term, a common choice adopted broadly in existing graph regularized sparse representation methods.

Chapter 4 describes similarity learning from the perspective of learning the subspace structures of the high dimensional data. We propose ℓ^0-graph, which generates subspace consistent sparse graph for clustering and semi-supervised learning. Sparse subspace learning methods, such as Sparse Subspace Clustering (SSC), assume that the high dimensional data lie in a union of subspaces, and they aim to build a sparse graph where a data point is only connected to other data that lie in the same subspace. Such sparse graph is called subspace consistent sparse graph. Data belonging to different subspaces are disconnected in the subspace consistent sparse graph; therefore, compelling clustering and semi-supervised learning performance are achieved by applying standard graph based machine learning methods, such as spectral clustering [10] and label propagation

[11], over the subspace-consistent sparse graph. Most of sparse subspace clustering methods require certain assumptions, e.g. independence or disjointness, on the subspaces to obtain the subspace consistent sparse graph. These assumptions are not guaranteed to hold in practice and they limit the application of existing sparse subspace learning methods on subspaces with general location. In this dissertation, we propose ℓ^0-graph, which obtains the subspace-consistent sparse graph for arbitrary distinct underlying subspaces almost surely under the mild i.i.d. assumption on the data generation. Extensive experimental results on various data sets demonstrate the superiority of ℓ^0-graph over other methods including SSC for both clustering and semi-supervised learning.

The above two similarity modeling methods produce a sparse graph over the data using the entire data as the dictionary in the sparse approximation procedure. Therefore, their efficiency is hindered by the size of the data. In order to overcome this challenge, we propose Support Regularized Sparse Coding (SRSC) in Chapter 5, which learns a compact dictionary rather than using the entire data as the dictionary. In contrast to ordinary sparse coding where the sparse code with fixed dictionary is independent for each data point without considering the geometric information and manifold structure of the entire data, SRSC produces sparse code that accounts for the manifold structure of the data by encouraging nearby data in the manifold to choose similar dictionary atoms. In this way, the obtained support regularized sparse code captures the locally linear structure of the data manifold. The similarity of two data points is intuitively set to the positive part of the inner product of the corresponding support regularized sparse codes, leading to compelling clustering performance. Moreover, we design a novel feed-forward neural network named Deep Support Regularized Sparse Coding (Deep-SRSC) as a fast encoder to approximate the sparse code generated by SRSC. Instead of the ordinary optimization method which requires time-consuming numerous iterations, Deep-SRSC outputs the support regularized sparse codes by feeding the data into a network which is comprised of only a few layers, and the architecture of the layer is designed in an interpretable way according to the optimization of SRSC. Experimental results on real data demonstrate the effectiveness of Deep-SRSC.

CHAPTER 2

ON A THEORY OF NONPARAMETRIC PAIRWISE SIMILARITY FOR CLUSTERING: CONNECTING CLUSTERING TO CLASSIFICATION

2.1 Introduction

Pairwise clustering methods partition the data into a set of self-similar clusters based on the pairwise similarity between the data points. Representative clustering methods include K-means [12] which minimizes the within-cluster dissimilarities, spectral clustering [10] which identifies clusters of more complex shapes lying on low dimensional manifolds, and the pairwise clustering method [13] using message-passing algorithm to inference the cluster labels in a pairwise undirected graphical model. Utilizing pairwise similarity, these pairwise clustering methods often avoid estimating complex hidden variables or parameters, which is difficult for high dimensional data.

However, most pairwise clustering methods assume that the pairwise similarity is given [12, 10], or they learn a more complicated similarity measure based on several given base similarities [13]. In this chapter, we present a new framework for pairwise clustering where the pairwise similarity is derived as the generalization error bound for the unsupervised nonparametric classifier. The unsupervised classifier is learned from unlabeled data and the hypothetical labeling. The quality of the hypothetical labeling is measured by the associated generalization error of the learned classifier, and the hypothetical labeling with minimum associated generalization error bound is preferred. We consider two nonparametric classifiers, i.e. the nearest neighbor classifier (NN) and the plug-in classifier (or the kernel density classifier). The generalization error bounds for both unsupervised classifiers are expressed as sum of pairwise terms between the data points, which can be interpreted as a nonparametric pairwise similarity measure between the data points. Under uniform distribution, both nonparametric similarity measures exhibit a well-known form of kernel similarity. We also prove that the generalization error bound for the unsupervised plug-in classifier is asymptotically equal

to the weighted volume of cluster boundary [1] for Low Density Separation, a widely used criterion for semi-supervised learning and clustering.

Our work is closely related to discriminative clustering methods by unsupervised classification, which searches for the cluster boundaries with the help of an unsupervised classifier. For example, [14] learns a max-margin two-class classifier to group unlabeled data in an unsupervised manner, known as unsupervised SVM, whose theoretical property is further analyzed in [15]. Also, [16] learns the kernel logistic regression classifier, and uses the entropy of the posterior distribution of the class label by the classifier to measure the quality of the learned classifier. More recent work presented in [17] learns an unsupervised classifier by maximizing the mutual information between cluster labels and the data, and the Squared-Loss Mutual Information is employed to produce a convex optimization problem. Although such discriminative methods produce satisfactory empirical results, the optimization of complex parameters hampers their application in high-dimensional data. Following the same principle of unsupervised classification using nonparametric classifiers, we derive nonparametric pairwise similarity and eliminate the need of estimating complicated parameters of the unsupervised classifier. As an application, we develop a new nonparametric exemplar-based clustering method with the derived nonparametric pairwise similarity induced by the plug-in classifier, and our new method demonstrates better empirical clustering results than the existing exemplar-based clustering methods.

It should be emphasized that our generalization bounds are essentially different from the literature. As nonparametric classification methods, the generalization properties of the nearest neighbor classifier (NN) and the plug-in classifier are extensively studied. Previous research focuses on the average generalization error of the NN [18, 19], which is the average error of the NN over all the random training data sets, or the excess risk of the plug-in classifier [20, 21]. In [18], it is shown that the average generalization error of the NN is bounded by twice the Bayes error. Assuming that the class of the regression functions has a smooth parameter β, [20] proves that the excess risk of the plug-in classifier converges to 0 of the order $n^{\frac{-\beta}{2\beta+d}}$ where d is the dimension of the data. [21] further shows that the plug-in classifier attains faster convergence rate of the excess risk, namely $n^{-\frac{1}{2}}$, under some margin assumption on the data distribution. All these generalization error bounds depend on the unknown Bayes error. By virtue of kernel density estimation and generalized kernel density estimation [22], our generalization bounds are represented mostly in terms of the data, leading to the pairwise similarities for

clustering.

2.2 Formulation of Pairwise Clustering by Unsupervised Nonparametric Classification

The discriminative clustering literature [14, 16] has demonstrated the potential of multi-class classification for the clustering problem. Inspired by the natural connection between clustering and classification, we model the clustering problem as a multi-class classification problem: a classifier is learned from the training data built by a hypothetical labeling, which is a possible cluster labeling. The optimal hypothetical labeling is supposed to be the one such that its associated classifier has the minimum generalization error bound. To study the generalization bound for the classifier learned from the hypothetical labeling, we define the concept of classification model. Given unlabeled data $\{\mathbf{x}_l\}_{l=1}^n$, a classification model $M_{\mathcal{Y}}$ is constructed for any hypothetical labeling $\mathcal{Y} = \{\mathbf{y}_l\}_{l=1}^n$ as below:

Definition 1. *The classification model corresponding to the hypothetical labeling $\mathcal{Y} = \{\mathbf{y}_l\}_{l=1}^n$ is defined as $M_{\mathcal{Y}} = \left(\mathcal{S}, P_{XY}, \{\pi_i, f_i\}_{i=1}^Q, F\right)$. $\mathcal{S} = \{\mathbf{x}_l, \mathbf{y}_l\}_{l=1}^n$ are the labeled data by the hypothetical labeling, and $\mathcal{S}$ are assumed to be i.i.d. samples drawn from the joint distribution $P_{XY} = P_{X|Y}P_Y$, where (X, Y) is a random couple, $X \in \mathbb{R}^d$ represents the data and $Y \in \{1, 2, ..., Q\}$ is the class label of X, Q is the number of classes determined by the hypothetical labeling. Furthermore, P_{XY} is specified by $\{\pi^{(i)}, f^{(i)}\}_{i=1}^Q$ as follows: $\pi^{(i)}$ is the class prior for class i, i.e. $\Pr[Y = i] = \pi^{(i)}$; the conditional distribution $P_{X|Y=i}$ has probabilistic density function $f^{(i)}$, $i = 1, \ldots, Q$. F is a classifier trained using the training data $\mathcal{S}$. The generalization error of the classification model $M_{\mathcal{Y}}$ is defined as the generalization error of the classifier F in $M_{\mathcal{Y}}$.*

In this chapter, we study two types of classification models with the nearest neighbor classifier and the plug-in classifier respectively, and derive their generalization error bounds as sum of pairwise similarity between the data. Given a specific type of classification model, the optimal hypothetical labeling corresponds to the classification model with minimum generalization error bound. The optimal hypothetical labeling also generates a data partition where the sum of pairwise similarity between the data from different clusters is minimized, which is a common criterion for discriminative clustering.

In the following text, we derive the generalization error bounds for the two types of classification models. Before that, we introduce more notations and assumptions for the classification model. Denote by P_X the induced marginal distribution of X, and f is the probabilistic density function of P_X which is a mixture of Q class-conditional densities: $f = \sum_{i=1}^{Q} \pi^{(i)} f^{(i)}$. $\eta^{(i)}(x)$ is the regression function of Y on $X = x$, i.e. $\eta^{(i)}(x) = \Pr[Y = i \,|X = x] = \frac{\pi^{(i)} f^{(i)}(x)}{f(x)}$. For the sake of the consistency of the kernel density estimators used in the sequel, there are further assumptions on the marginal density and class-conditional densities in the classification model for any hypothetical labeling:
(A) f is bounded from below, i.e. $f \geq f_{\min} > 0$.
(B) $\{f^{(i)}\}$ is bounded from above, i.e. $f^{(i)} \leq f^{(i)}_{\max}$, and $f^{(i)} \in \Sigma_{\gamma,c_i}$, $1 \leq i \leq Q$.
where $\Sigma_{\gamma,c}$ is the class of Hölder-γ smooth functions with Hölder constant c:

$$\Sigma_{\gamma,c} \triangleq \{f : \mathbb{R}^d \to \mathbb{R} \,|\, \forall x, y, |f(x) - f(y)| \leq c\|x - y\|^{\gamma}\}, \gamma > 0$$

It follows from assumption (B) that $f \in \Sigma_{\gamma,c}$ where $c = \sum_i \pi^{(i)} c_i$. Assumptions (A) and (B) are mild. The upper bound for the density functions is widely required for the consistency of kernel density estimators [23, 24]; Hölder-γ smoothness is required to bound the bias of such estimators, and it also appears in [21] for estimating the excess risk of the plug-in classifier. The lower bound for the marginal density is used to derive the consistency of the estimator of the regression function $\eta^{(i)}$ (Lemma 2) and the consistency of the generalized kernel density estimator (Lemma 3). We denote by $\mathcal{P}_X$ the collection of marginal distributions that satisfy assumption (A), and denote by $\mathcal{P}_{X|Y}$ the collection of class-conditional distributions that satisfy assumption (B). We then define the collection of joint distributions $\mathcal{P}_{XY}$ that P_{XY} belongs to, which requires the marginal density and class-conditional densities satisfy assumption (A)-(B):

$$\mathcal{P}_{XY} \triangleq \{P_{XY} \mid P_X \in \mathcal{P}_X, \{P_{X|Y=i}\} \in \mathcal{P}_{X|Y}, \min_i\{\pi^{(i)}\} > 0\} \tag{2.1}$$

Given the joint distribution P_{XY}, the generalization error of the classifier F learned from the training data $\mathcal{S}$ is:

$$R(F_{\mathcal{S}}) \triangleq \Pr[(X, Y) : F(X) \neq Y] \tag{2.2}$$

Nonparametric kernel density estimator (KDE) serves as the primary tool of esti-

mating the underlying probabilistic density functions in our generalization analysis, and we introduce the KDE of f as follows:

$$\hat{f}_{n,h_n}(x) = \frac{1}{n}\sum_{l=1}^{n} K_{h_n}(x - \mathbf{x}_l) \tag{2.3}$$

where $K_h(x) = \frac{1}{h^d}K\left(\frac{x}{h}\right)$ is the isotropic Gaussian kernel with bandwidth h and $K(x) \triangleq \frac{1}{(2\pi)^{d/2}}e^{-\frac{\|x\|^2}{2}}$. We have the following VC property of the Gaussian kernel K. Define the class of functions

$$\mathcal{F} \triangleq \{K\left(\frac{t-\cdot}{h}\right), t \in \mathbb{R}^d, h \neq 0\} \tag{2.4}$$

The VC property appears in [23, 24, 25, 26, 27], and it is proved that $\mathcal{F}$ is a bounded VC class of measurable functions with respect to the envelope function F such that $|u| \leq F$ for any $u \in \mathcal{F}$ (e.g. $F \equiv (2\pi)^{-\frac{d}{2}}$). It follows that there exist positive numbers A and v such that for every probability measure P on $\mathbb{R}^d$ for which $\int F^2 dP < \infty$ and any $0 < \tau < 1$,

$$N\left(\mathcal{F}, \|\cdot\|_{L_2(P)}, \tau\|F\|_{L_2(P)}\right) \leq \left(\frac{A}{\tau}\right)^v \tag{2.5}$$

where $N\left(\mathcal{T}, \hat{d}, \epsilon\right)$ is defined as the minimal number of open $\hat{d}$-balls of radius ϵ required to cover $\mathcal{T}$ in the metric space $\left(\mathcal{T}, \hat{d}\right)$. A and v are called the VC characteristics of $\mathcal{F}$.

The VC property of K is required for the consistency of kernel density estimators shown in Lemma 2. Also, we adopt the following kernel estimator of $\eta^{(i)}$:

$$\hat{\eta}^{(i)}_{n,h_n}(x) = \frac{\sum_{l=1}^{n} K_{h_n}(x - \mathbf{x}_l)\mathbb{I}_{\{\mathbf{y}_l = i\}}}{n\hat{f}_{n,h_n}(x)} \tag{2.6}$$

Before stating Lemma 2, we introduce several frequently used quantities throughout this chapter. Let $L, C > 0$ be constants which only depend on the VC characteristics of the Gaussian kernel K. We define

$$f_0 \triangleq \sum_{i=1}^{Q} \pi^{(i)} f^{(i)}_{\max} \quad \sigma_0^2 \triangleq \|K\|_2^2 f_0 \tag{2.7}$$

Also, for all positive numbers $\lambda \geq C$ and $\sigma > 0$, we define

$$E_{\sigma^2} \triangleq \frac{\log\left(1 + \lambda/4L\right)}{\lambda L \sigma^2} \tag{2.8}$$

Based on Corollary 2.2 in [23], Lemma 2 and Lemma 3 in Section 2.6 show the strong consistency (almost sure uniformly convergence) of several kernel density estimators, i.e. $\hat{f}_{n,h_n}$, $\{\hat{\eta}^{(i)}_{n,h_n}\}$ and the generalized kernel density estimator, and they form the basis for the derivation of the generalization error bounds for the two types of classification models.

2.3 Generalization Bounds

We derive the generalization error bounds for the two types of classification models with the nearest neighbor classifier and the plug-in classifier respectively. Substituting these kernel density estimators for the corresponding true density functions, Theorem 1 and 2 present the generalization error bounds for the classification models with the plug-in classifier and the nearest neighbor classifier. The dominant terms of both bounds are expressed as sum of pairwise similarity depending solely on the data, which facilitates the application of clustering. We also show the connection between the error bound for the plug-in classifier and Low Density Separation in this section. The detailed proofs are included in the supplementary.

2.3.1 Generalization Bound for the Classification Model with Plug-In Classifier

The plug-in classifier resembles the Bayes classifier while it uses the kernel density estimator of the regression function $\eta^{(i)}$ instead of the true $\eta^{(i)}$. It has the form

$$\mathrm{PI}\left(X\right) = \underset{1 \leq i \leq Q}{\arg\max}\, \hat{\eta}^{(i)}_{n,h_n}\left(X\right) \tag{2.9}$$

where $\hat{\eta}^{(i)}_{n,h_n}$ is the nonparametric kernel estimator of the regression function $\eta^{(i)}$

by (2.6). The generalization capability of the plug-in classifier has been studied by the literature[20, 21]. Letting F^* be the Bayes classifier, it is proved that the excess risk of PI_S, namely $\mathbb{E}_S R\left(\mathrm{PI}_S\right) - R\left(F^*\right)$, converges to 0 of the order $n^{\frac{-\beta}{2\beta+d}}$ under some complexity assumption on the class of the regression functions with smooth parameter β that $\{\eta^{(i)}\}$ belongs to [20, 21]. However, this result cannot be used to derive the generalization error bound for the plug-in classifier comprising of nonparametric pairwise similarities in our setting.

We show the upper bound for the generalization error of PI_S in Lemma 1.

Lemma 1. *For any $P_{XY} \in \mathcal{P}_{XY}$, there exists a n_0 which depends on σ_0 and VC characteristics of K such that when $n > n_0$, with probability greater than $1 - 2QLh_n^{E_{\sigma_0^2}}$, the generalization error of the plug-in classifier satisfies*

$$R\left(\mathrm{PI}_S\right) \leq R_n^{\mathrm{PI}} + \mathcal{O}\Big(\sqrt{\frac{\log h_n^{-1}}{nh_n^d}} + h_n^\gamma\Big) \tag{2.10}$$

$$R_n^{\mathrm{PI}} = \sum_{i,j=1,\ldots,Q, i\neq j} \mathbb{E}_X\left[\hat{\eta}_{n,h_n}^{(i)}\left(X\right)\hat{\eta}_{n,h_n}^{(j)}\left(X\right)\right] \tag{2.11}$$

where E_{σ^2} is defined by (A.2), h_n is chosen such that $h_n \to 0$, $\frac{\log h_n^{-1}}{nh_n^d} \to 0$, $\hat{\eta}_{n,h_n}^{(i)}$ is the kernel estimator of the regression function. Moreover, the equality in (A.3) holds when $\hat{\eta}_{n,h_n}^{(i)} \equiv \frac{1}{Q}$ for $1 \leq i \leq Q$.

Based on Lemma 1, we can bound the error of the plug-in classifier from above by R_n^{PI}. Theorem 1 then gives the bound for the error of the plug-in classifier in the corresponding classification model using the generalized kernel density estimator in Lemma 3. The bound has a form of sum of pairwise similarity between the data from different classes.

Theorem 1. *(Error of the Plug-In Classifier) Given the classification model $M_{\mathcal{Y}} = \Big(\mathcal{S}, P_{XY}, \{\pi_i, f_i\}_{i=1}^Q, \mathrm{PI}\Big)$ with $P_{XY} \in \mathcal{P}_{XY}$, there exists a n_1 which depends on σ_0, σ_1 and the VC characteristics of K such that when $n > n_1$, with probability greater than $1 - 2QLh_n^{E_{\sigma_0^2}} - L(\sqrt{2}h_n)^{E_{\sigma_0^2}} - QLh_n^{E_{\sigma_1^2}}$, the generalization error of the plug-in classifier satisfies*

$$R\left(\mathrm{PI}_S\right) \leq \hat{R}_n\left(\mathrm{PI}_S\right) + \mathcal{O}\Big(\sqrt{\frac{\log h_n^{-1}}{nh_n^d}} + h_n^\gamma\Big) \tag{2.12}$$

where $\hat{R}_n\left(\mathrm{PI}_S\right) = \frac{1}{n^2}\sum_{l,m}\theta_{lm}G_{lm,\sqrt{2}h_n}$, $\sigma_1^2 = \frac{\|K\|_2^2 f_{\max}}{f_{\min}}$, $\theta_{lm} = \mathbb{1}_{\{\mathbf{y}_l \neq \mathbf{y}_m\}}$ *is a class indicator function and*

$$G_{lm,h} = G_h\left(\mathbf{x}_l, \mathbf{x}_m\right),\ G_h\left(x,y\right) = \frac{K_h\left(x-y\right)}{\hat{f}_{n,h}^{\frac{1}{2}}\left(x\right)\hat{f}_{n,h}^{\frac{1}{2}}\left(y\right)} \tag{2.13}$$

E_{σ^2} *is defined by (A.2),* h_n *is chosen such that* $h_n \to 0$, $\frac{\log h_n^{-1}}{nh_n^d} \to 0$, $\hat{f}_{n,h_n}$ *is the kernel density estimator of* f *defined by (2.3).*

$\hat{R}_n$ is the dominant term determined solely by the data and the excess error $\mathcal{O}\left(\sqrt{\frac{\log h_n^{-1}}{nh_n^d}} + h_n^\gamma\right)$ goes to 0 with infinite n. In the following subsection, we show the close connection between the error bound for the plug-in classifier and the weighted volume of cluster boundary, and the latter is proposed by [1] for Low Density Separation.

Connection to Low Density Separation

Low Density Separation [28], a well-known criterion for clustering, requires that the cluster boundary should pass through regions of low density. It has been extensively studied in unsupervised learning and semi-supervised learning [29, 30, 31]. Suppose the data $\{\mathbf{x}_l\}_{l=1}^n$ lies on a domain $\Omega \subseteq R^d$. Let f be the probability density function on Ω, S be the cluster boundary which separates Ω into two parts S_1 and S_2. Following the Low Density Separation assumption, [1] suggests that the cluster boundary S with low weighted volume $\int_S f\left(s\right)ds$ should be preferable. [1] also proves that a particular type of cut function converges to the weighted volume of S. Based on their study, we obtain the following result relating the error of the plug-in classifier to the weighted volume of the cluster boundary.

Corollary 1. *Under the assumption of Theorem 1, for any kernel bandwidth sequence* $\{h_n\}_{n=1}^\infty$ *such that* $\lim_{n\to\infty} h_n = 0$ *and* $h_n > n^{-\alpha}$ *where* $0 < \alpha < \frac{1}{2d+2}$*, with probability* 1*,*

$$\lim_{n\to\infty}\frac{\sqrt{\pi}}{2h_n}\hat{R}_n\left(\mathrm{PI}_S\right) = \int_S f\left(s\right)ds \tag{2.14}$$

2.3.2 Generalization Bound for the Classification Model with Nearest Neighbor Classifier

Theorem 2 shows the generalization error bound for the classification model with nearest neighbor classifier (NN), which has a form similar to that of (A.5).

Theorem 2. *(Error of the NN)*
Suppose the classification model is given as $M_{\mathcal{Y}} = \left(\mathcal{S}, P_{XY}, \{\pi_i, f_i\}_{i=1}^{Q}, \mathrm{NN}\right)$ *with* $P_{XY} \in \mathcal{P}_{XY}$ *and the support of* P_X *is bounded by* $[-M_0, M_0]^d$*, there exists a* n_0 *which depends on* σ_0 *and VC characteristics of* K *such that when* $n > n_0$*, with probability greater than* $1 - 2QLh_n^{E_{\sigma_0^2}} - (2M_0)^d n^{dd_0} e^{-n^{1-dd_0} f_{\min}}$*, the generalization error of the NN satisfies:*

$$R\left(\mathrm{NN}_S\right) \leq \hat{R}_n\left(\mathrm{NN}_S\right) + c_0 \left(\sqrt{d}\right)^{\gamma} n^{-d_0\gamma} + \mathcal{O}\Big(\sqrt{\frac{\log h_n^{-1}}{n h_n^d}} + h_n^{\gamma}\Big) \tag{2.15}$$

where $\hat{R}_n\left(\mathrm{NN}\right) = \frac{1}{n} \sum_{1 \leq l < m \leq n} H_{lm,h_n} \theta_{lm}$*,*

$$H_{lm,h_n} = K_{h_n}\left(\mathbf{x}_l - \mathbf{x}_m\right) \Big(\frac{\int_{\mathcal{V}_l} \hat{f}_{n,h_n}(x)\,\mathrm{d}x}{\hat{f}_{n,h_n}\left(\mathbf{x}_l\right)} + \frac{\int_{\mathcal{V}_m} \hat{f}_{n,h_n}(x)\,\mathrm{d}x}{\hat{f}_{n,h_n}\left(\mathbf{x}_m\right)} \Big) \tag{2.16}$$

E_{σ^2} *is defined by (A.2),* d_0 *is a constant such that* $dd_0 < 1$*,* $\hat{f}_{n,h_n}$ *is the kernel density estimator of* f *defined by (2.3) with the kernel bandwidth* h_n *satisfying* $h_n \to 0, \frac{\log h_n^{-1}}{n h_n^d} \to 0$*,* $\mathcal{V}_l$ *is the Voronoi cell associated with* $\mathbf{x}_l$*,* c_0 *is a constant,* $\theta_{lm} = \mathbb{1}_{\{\mathbf{y}_l \neq \mathbf{y}_m\}}$ *is a class indicator function such that* $\theta_{lm} = 1$ *if* $\mathbf{x}_l$ *and* $\mathbf{x}_m$ *belongs to different classes, and* 0 *otherwise. Moreover, the equality in (A.8) holds when* $\eta^{(i)} \equiv \frac{1}{Q}$ *for* $1 \leq i \leq Q$*.*

$G_{lm,\sqrt{2}h_n}$ in (A.6) and H_{lm,h_n} in (A.9) are the new pairwise similarity functions induced by the plug-in classifier and the nearest neighbor classifier respectively. According to the proof of Theorem 1 and Theorem 2, the kernel density estimator $\hat{f}$ can be replaced by the true density f in the denominators of (A.6) and (A.9), and the conclusions of Theorem 1 and 2 still hold. Therefore, both $G_{lm,\sqrt{2}h_n}$ and H_{lm,h_n} are equal to ordinary Gaussian kernels (up to a scale) with different kernel bandwidth under uniform distribution, which explains the broadly used kernel similarity in data clustering from an angle of supervised learning.

2.4 Application to Exemplar-Based Clustering

We propose a nonparametric exemplar-based clustering algorithm using the derived nonparametric pairwise similarity by the plug-in classifier. In exemplar-based clustering, each $\mathbf{x}_l$ is associated with a cluster indicator e_l ($l \in \{1, 2, ...n\}$, $e_l \in \{1, 2, ...n\}$), indicating that $\mathbf{x}_l$ takes $\mathbf{x}_{e_l}$ as the cluster exemplar. Data from the same cluster share the same cluster exemplar. We define $\mathbf{e} \triangleq \{e_l\}_{l=1}^n$. Moreover, a configuration of the cluster indicators $\mathbf{e}$ is consistent iff $e_l = l$ when $e_m = l$ for any $l, m \in 1..n$, meaning that $\mathbf{x}_l$ should take itself as its exemplar if any $\mathbf{x}_m$ take $\mathbf{x}_l$ as its exemplar. It is required that the cluster indicators $\mathbf{e}$ should always be consistent. Affinity Propagation (AP) [32], a representative of the exemplar-based clustering methods, solves the following optimization problem:

$$\min_{\mathbf{e}} \sum_{l=1}^{n} S_{l,e_l} \quad s.t. \quad \mathbf{e} \text{ is consistent} \tag{2.17}$$

S_{l,e_l} is the dissimilarity between x_l and x_{e_l}, and note that $S_{l,l}$ is set to be nonzero to avoid the trivial minimizer of (2.17).

Now we aim to improve the discriminative capability of the exemplar-based clustering (2.17) using the nonparametric pairwise similarity derived by the unsupervised plug-in classifier. As mentioned before, the quality of the hypothetical labeling $\hat{\mathbf{y}}$ is evaluated by the generalization error bound for the nonparametric plug-in classifier trained by $S_{\hat{\mathbf{y}}}$, and the hypothetical labeling $\hat{\mathbf{y}}$ with minimum associated error bound is preferred, i.e. $\arg\min_{\hat{\mathbf{y}}} \hat{R}_n\left(\mathrm{PI}_S\right) = \arg\min_{\hat{\mathbf{y}}} \sum_{l,m} \theta_{lm} G_{lm,\sqrt{2}h_n}$ where $\theta_{lm} = \mathbb{1}_{\hat{y}_l \neq \hat{y}_m}$ and $G_{lm,\sqrt{2}h_n}$ is defined in (A.6). By Lemma 3, minimizing $\sum_{l,m} \theta_{lm} G_{lm,\sqrt{2}h_n}$ also enforces minimization of the weighted volume of cluster boundary asymptotically. To avoid the trivial clustering where all the data are grouped into a single cluster, we use the sum of within-cluster dissimilarities term $\sum_{l=1}^{n} \exp\left(-G_{le_l,\sqrt{2}h_n}\right)$ to control the size of clusters. Therefore, the objective function of our pairwise clustering method is

$$\Psi(e) = \sum_{l=1}^{n} \exp\left(-G_{le_l,\sqrt{2}h_n}\right) + \lambda \sum_{l,m} \left(\tilde{\theta}_{lm} G_{lm,\sqrt{2}h_n} + \rho_{lm}(e_l, e_m)\right) \tag{2.18}$$

where ρ_{lm} is a function to enforce the consistency of the cluster indicators:

$$\rho_{lm}(e_l, e_m) = \begin{cases} \infty & e_m = l, e_l \neq l \text{ or } e_l = m, e_m \neq m \\ 0 & \text{otherwise} \end{cases}$$

and λ is a balancing parameter. Due to the form of (A.36), we construct a pairwise Markov Random Field (MRF) representing the unary term u_l and the pairwise term $\tilde{\theta}_{lm}G_{lm,\sqrt{2}h_n} + \rho_{lm}$ as the data likelihood and prior respectively. The variables $\mathbf{e}$ are modeled as nodes and the unary term and pairwise term in (A.36) are modeled as potential functions in the pairwise MRF. The minimization of the objective function is then converted to a MAP (Maximum a Posterior) problem in the pairwise MRF. (A.36) is minimized by Max-Product Belief Propagation (BP).

The computational complexity of our clustering algorithm is $\mathcal{O}(TEN)$, where E is the number of edges in the pairwise MRF, T is the number of iterations of message passing in the BP algorithm. We call our new algorithm Plug-In Exemplar Clustering (PIEC), and compare it to representative exemplar-based clustering methods, i.e. AP and Convex Clustering with Exemplar-Based Model (CEB) [33], for clustering on three real data sets from the UCI repository, i.e. Iris, Vertebral Column (VC) and Breast Tissue (BT). We record the average clustering accuracy (AC) and the standard deviation of AC for all the exemplar-based clustering methods when they produce the correct number of clusters for each data set with different values of h_n and λ, and the results are shown in Table 2.1. Although AP produces better clustering accuracy on the VC data set, PIEC generates the correct cluster numbers more often. The default value for the kernel bandwidth h_n is h_n^*, which is set as the variance of the pairwise distance between data points $\left\{\|\mathbf{x}_l - \mathbf{x}_m\|_{l<m}\right\}$. The default value for the balancing parameter λ is 1. We let $h_n = \alpha h_n^*$, λ varies between $[0.2, 1]$ and α varies between $[0.2, 1.9]$ with step 0.2 and 0.05 respectively, resulting in 170 different parameter settings. We also generate the same number of parameter settings for AP and CEB.

Table 2.1: Comparison between exemplar-based clustering methods. The number in the bracket is the number of times when the corresponding algorithm produces correct cluster numbers.

Data sets	Iris	VC	BT
AP	0.8933 ± 0.0138 (16)	**0.6677** (14)	0.4906 (1)
CEB	0.6929 ± 0.0168 (15)	0.4748 ± 0.0014 (5)	0.3868 ± 0.08 (2)
PIEC	**0.9089** $\pm$ **0.0033** (15)	0.5263 ± 0.0173 (35)	**0.6585** ± 0.0103 (5)

2.5 Conclusion

We propose a new pairwise clustering framework where nonparametric pairwise similarity is derived by minimizing the generalization error unsupervised nonparametric classifier. Our framework bridges the gap between clustering and multiclass classification, and explains the widely used kernel similarity for clustering. In addition, we prove that the generalization error bound for the unsupervised plug-in classifier is asymptotically equal to the weighted volume of cluster boundary for Low Density Separation. Based on the derived nonparametric pairwise similarity using the plug-in classifier, we propose a new nonparametric exemplar-based clustering method with enhanced discriminative capability compared to the exiting exemplar-based clustering methods.

2.6 Consistency of Kernel Density Estimator and the Generalized Kernel Density Estimator

Lemma 2. *(Consistency of Kernel Density Estimator) Let the kernel bandwidth h_n of the Gaussian kernel K be chosen such that $h_n \to 0$, $\frac{\log h_n^{-1}}{nh_n^d} \to 0$. For any $P_X \in \mathcal{P}_X$, there exists a n_0 which depends on σ_0 and VC characteristics of K, when $n > n_0$, with probability greater than $1 - Lh_n^{E_{\sigma_0^2}}$ over the data $\{\mathbf{x}_l\}$,*

$$\left\| \hat{f}_{n,h_n}(x) - f(x) \right\|_\infty = \mathcal{O}\Big(\sqrt{\frac{\log h_n^{-1}}{nh_n^d}} + h_n^\gamma \Big) \tag{2.19}$$

where $\hat{f}_{n,h_n}$ is the kernel density estimator of f. Furthermore, for any $P_{XY} \in \mathcal{P}_{XY}$, when $n > n_0$, then with probability greater than $1 - 2Lh_n^{E_{\sigma_0^2}}$ over the data $\{\mathbf{x}_l\}$,

$$\left\| \hat{\eta}_{n,h_n}^{(i)}(x) - \eta^{(i)}(x) \right\|_\infty = \mathcal{O}\Big(\sqrt{\frac{\log h_n^{-1}}{nh_n^d}} + h_n^\gamma \Big) \tag{2.20}$$

for each $1 \le i \le Q$.

Lemma 3. *(Consistency of the Generalized Kernel Density Estimator) Suppose f is the probabilistic density function of $P_X \in \mathcal{P}_X$, and $f \le f_{\max}$. Let g be a bounded function defined on $\mathcal{X}$ and $g \in \Sigma_{\gamma,g_0}$, $0 < g_{\min} \le g \le g_{\max}$, and $e = \frac{f}{g}$.*

Define the generalized kernel density estimator of e *as*

$$\hat{e}_{n,h} \triangleq \frac{1}{n}\sum_{l=1}^{n}\frac{K_h\left(x-\mathbf{x}_l\right)}{g\left(\mathbf{x}_l\right)} \tag{2.21}$$

Let $\sigma_g^2 = \frac{\|K\|_2^2 f_{\max}}{g_{\min}^2}$. *There exists* n_g *which depends on* σ_g *and the VC characteristics of* K *such that when* $n > n_g$, *with probability greater than* $1 - Lh_n^{E_{\sigma_g^2}}$ *over the data* $\{\mathbf{x}_l\}$,

$$\left\|\hat{e}_{n,h_n}\left(x\right) - e\left(x\right)\right\|_\infty = \mathcal{O}\Big(\sqrt{\frac{\log h_n^{-1}}{nh_n^d}} + h_n^\gamma\Big) \tag{2.22}$$

where h_n *is chosen such that* $h_n \to 0$, $\frac{\log h_n^{-1}}{nh_n^d} \to 0$.

Sketch of proof: For fixed $h \neq 0$, we consider the class of functions

$$\mathcal{F}_g \triangleq \{\frac{K\left(\frac{t-\cdot}{h}\right)}{g\left(\cdot\right)}, t \in \mathbb{R}^d, h \neq 0\}$$

It can be verified that $\mathcal{F}_g$ is also a bounded VC class with the envelope function $F_g = \frac{F}{g_{\min}}$, and

$$N\left(\mathcal{F}_g, \left\|\cdot\right\|_{L_2(P)}, \tau\left\|F_g\right\|_{L_2(P)}\right) \leq \left(\frac{A}{\tau}\right)^v \tag{2.23}$$

Then (A.13) follows from similar argument in the proof of Lemma 2 and Corollary 2.2 in [23]. □

The generalized kernel density estimator (A.12) is also used in [22] to estimate the Laplacian PDF Distance between two probabilistic density functions, and the authors only provide the proof of pointwise weak consistency of this estimator in [22]. Under mild conditions, our Lemma 2 and 3 show the strong consistency of the generalized kernel density estimator and the traditional kernel density estimator under the same theoretical framework of the VC property of the kernel.

CHAPTER 3

MANIFOLD LEARNING WITH ℓ^0 REGULARIZED ℓ^1-GRAPH

3.1 Introduction

Clustering is a common unsupervised data analysis method which partitions data into a set of self-similar clusters. The data clusters always serve as an indispensable prerequisite for solving various machine learning and computer vision problems by the disclosed grouping patters in the original data. Most clustering algorithms are either similarity-based or model-based methods. Model-based clustering methods typically model the data distribution statistically, for example, by a mixture of parametric distributions [34]. The parameters of the model are estimated via fitting a statistical model to the data. However, difficulty on the parameter estimation imposed by high dimensionality always hinders the application of model-based methods; even in the case that parameter estimation is feasible, the resultant parametric distribution is not guaranteed to match the underlying true distribution of the data, especially in the case of high dimensionality, which further restricts the feasibility of model-based methods.

In contrast, similarity-based clustering methods segment the data based on the similarity measure between the data points, so they avoid the difficult problem of parameter estimation. For example, K-means [35] searches for data clusters by a local minimum of sum of within-cluster dissimilarities. Affinity Propagation (AP) [36] uses the same principle and it automatically determines the cluster number, and [37] further improves its discriminative capability for pairwise clustering. Spectral Clustering [10] identifies clusters of complex shapes lying on some low dimensional manifolds by spectral embedding. Among various similarity-based clustering methods, graph-based methods [38] are important, wherein the data similarity is the edge weight of the graph, and sparse graph which has only a few edges of nonzero weights for each vertex is demonstrated to be effective, especially for clustering high dimensional data. Examples of sparse graph based

clustering methods include ℓ^1-graph [2, 3] and Sparse Subspace Clustering (SSC) [4], which build the sparse graph by reconstructing each datum with all the other data by sparse representation. In the sparse graph produced by ℓ^1-graph or SSC, the vertices represent the data, and an edge is between two vertices whenever one participates in the spare representation of the other. The weight of the edge is the average of the associated elements in the sparse codes corresponding to the two vertices. A theoretical explanation is provided by SSC, which shows that such sparse representation recovers the underlying subspaces from which the data are drawn under certain assumptions, such as the independence or disjointness assumption on the subspaces. When such assumptions hold, data belonging to different subspaces are disconnected in the sparse graph. A sparse similarity matrix is then obtained as the weighted adjacency matrix of the constructed sparse graph by ℓ^1-graph or SSC, and spectral clustering is performed on the sparse similarity matrix to obtain the data clusters. ℓ^1-graph and SSC have been shown to be robust to noise and capable of producing superior results for high dimensional data, compared to spectral clustering on the similarity produced by the widely used Gaussian kernel.

While ℓ^1-graph demonstrates compelling performance for clustering, it performs sparse representation for each datum independently without considering the geometric information of the data. High dimensional data always lie in low dimensional submanifold. In this chapter, manifold assumption [5] is employed to obtain the sparse representation complying to geometric information of the data so as to improve the performance of ℓ^1-graph. Manifold assumption [5], which assumes local smoothness of the embedding, has been employed in the literature of sparse representations to obtain the regularized sparse representations that take into account the geometric information and manifold structure of the data. Interpreting the sparse code of a data point as its embedding, the manifold assumption in most sparse representation methods requires that if two data points are close in the intrinsic geometry of the submanifold, their corresponding sparse codes are also expected to be similar to each other measured by ℓ^2-distance, or varying smoothly along the geometry of the submanifold. Graph Laplacian [39] is widely used to impose such local smoothness requirement on the sparse codes, and the associated graph regularization term mostly measures the distance between the sparse codes by ℓ^2-norm [8, 9]. In [8, 9], both dictionary and the regularized sparse code are learned, and the regularized sparse code is used as a feature representation of the corresponding data point in classification, clustering and other

learning tasks. However, in the context of sparse graph based clustering, the sparse graph determines the clustering performance. The information of each data point in the sparse graph is its associated local structure of the sparse graph, namely its neighbors and the associated edge weights. Therefore, to obtain a sparse graph in accordance with the manifold structure of the data, the local structure of the sparse graph for a data point serves as the embedding for that point, and manifold assumption in this context should impose local smoothness of the local structure of the sparse graph. Namely:

Manifold assumption on the local structure of the sparse graph: nearby data are encouraged to have similar local sparse graph structure, i.e. they should have similar neighbors and similar edge weights in the sparse graph.

In this chapter, we restrict the information of the local sparse graph structure for a data point to that contained in its sparse code, for the convenience of optimization in terms of the sparse codes. The support of the sparse code of a data point determines the neighbors it selects, and the nonzero elements of the sparse code contribute to the corresponding edge weights (please refer to the details of the sparse graph construction in Section 2). This also indicates that ℓ^2-distance is not a suitable distance measure for sparse codes in our setting, and one can easily imagine that two sparse codes can have very small ℓ^2-distance while their supports are quite different, meaning that they choose different neighbors. Motivated by the manifold assumption on the local sparse graph structure, we propose a novel ℓ^0 regularized ℓ^1-graph, abbreviated as ℓ^0-ℓ^1-graph, which uses ℓ^0-norm induced distance between the sparse codes to impose the local smoothness of the local sparse graph structure, leading to a new ℓ^0 graph regularization. Compared to ℓ^2-norm, ℓ^0-norm counts the number of nonzero elements in a vector and minimizing ℓ^0-norm induced distance between two sparse codes in our new regularization term encourages nearby data in the manifold to have similar local sparse graph structure in terms of their sparse codes. [1] Another benefit of the manifold assumption on the local sparse graph structure is that, instead of choosing neighbors by itself, each data point has to coordinate with its nearby data points in the manifold to choose its neighbors, which makes the constructed sparse graph more robust to outliers.

Although there are sparse representation methods [40, 41] that impose the spar-

[1] Note that for a data point, its nearby data points (in the manifold) are usually specified by a K-nearest-neighbor graph, as will be illustrated in Section 3.2, and they are different from its neighbors in the sparse graph for clustering.

sity of the sparse codes by ℓ^0-norm, to the best of our knowledge, ℓ^0-ℓ^1-graph is the first method that encourages the local smoothness of the sparse graph structure by ℓ^0-norm so as to render sparse graph complying to the intrinsic geometric structure of the data in the context of sparse graph based clustering. Although ℓ^0-norm is not continuous, previous sparse representation methods such as [40] that directly optimize objective function involving ℓ^0-norm demonstrate compelling performance. In addition, instead of the greedy method such as Orthogonal Matching Pursuit (OMP) [42], we develop a proximal method to optimize the nonconvex objective function of ℓ^0-ℓ^1-graph with convergence guarantee.

The remainder of the chapter is organized as follows. Sparse coding, ℓ^1-graph and ℓ^2-ℓ^1-graph are introduced in the next section, and then the detailed formulation of ℓ^0-ℓ^1-graph is illustrated. We then show the clustering performance of ℓ^0-ℓ^1-graph, and conclude the chapter. We use bold letters for matrices and vectors, and regular lower letter for scalars throughout this chapter. The bold letter with superscript indicates the corresponding column of a matrix, and the bold letter with subscript indicates the corresponding element of a matrix or vector. $\|\cdot\|_F$ and $\|\cdot\|_p$ denote the Frobenius norm and the ℓ^p-norm, and $\mathrm{diag}(\cdot)$ indicates the diagonal elements of a matrix. Also, ℓ^0-norm induced distance is abbreviated as ℓ^0-distance throughout this chapter.

3.2 Preliminaries: Sparse Coding, ℓ^1-Graph and Its ℓ^2 Graph Regularization — ℓ^2-ℓ^1-Graph

The aim of sparse coding is to represent an input vector by a linear combination of a few atoms of a dictionary which is usually over-complete, and the coefficients for the atoms are called sparse code. Sparse coding is widely applied in machine learning and signal processing, and sparse code is extensively used as a discriminative and robust feature representation with convincing performance for classification and clustering [43, 44, 45]. Suppose the data $\boldsymbol{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ lie in the d-dimensional Euclidean space $\mathbb{R}^d$, and the dictionary matrix is $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_p] \in \mathbb{R}^{d\times p}$ and each $\mathbf{d}_k\,(k = 1, \ldots, p)$ is the atom of the dictionary. The sparse coding method seeks the linear sparse representation with respect to the dictionary $\mathbf{D}$ for each vector $\mathbf{x}_i$ by solving the following convex optimization

problem:

$$\boldsymbol{\alpha}^i = \arg\min_{\boldsymbol{\alpha}^i} \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}^i\|_2^2 + \lambda||\boldsymbol{\alpha}^i||_1 \quad i = 1, \ldots, n \tag{3.1}$$

where λ is a weighting parameter for the sparsity of $\boldsymbol{\alpha}^i$.

ℓ^1-graph [2, 3] and SSC [46, 4] apply the idea of sparse coding where the data similarity is represented by the sparse codes. Given the data $\boldsymbol{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$, ℓ^1-graph and SSC solve the following optimization problem to obtain the sparse representation for each data point:

$$\min_{\boldsymbol{\alpha}^i} \|\boldsymbol{\alpha}^i\|_1 \quad s.t.\ \mathbf{x}_i = \boldsymbol{X}\boldsymbol{\alpha}^i \tag{3.2}$$

In SSC [46, 4], it is proved that the sparse representation (4.1) for each datum recovers the underlying subspaces from which the data are generated when the subspaces are independent or disjoint, and certain conditions on the geometric properties, such as the principle angle between different subspaces, hold. When these required assumptions hold, data belonging to different subspaces are disconnected in the sparse graph, leading to the success of the subspace clustering. In practice, however, one can often empirically try the same formulation to obtain satisfactory results even without checking the assumptions.

Allowing some tolerance for inexact representation and robustness to noise [47, 48], the following Lasso-type problem is solved instead of (4.1):

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \quad s.t.\ \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{\alpha}\|_F \leq \delta,\ \mathrm{diag}(\boldsymbol{\alpha}) = \mathbf{0}$$

which is equivalent to

$$\min_{\boldsymbol{\alpha}^i} \|\mathbf{x}_i - \boldsymbol{X}\boldsymbol{\alpha}^i\|_2^2 + \lambda_{\ell^1}\|\boldsymbol{\alpha}^i\|_1 \quad i = 1, \ldots, n \tag{3.3}$$

for some weighting parameter $\lambda_{\ell^1} > 0$, and $\boldsymbol{\alpha}^i \in \mathbb{R}^{n\times 1}$, $\boldsymbol{\alpha} = [\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^n] \in \mathbb{R}^{n\times n}$ is the coefficient matrix with the element $\boldsymbol{\alpha}_{ij} = \boldsymbol{\alpha}_i^j$. The diagonal elements of $\boldsymbol{\alpha}$ are enforced to be zero, i.e. $\boldsymbol{\alpha}_{ii} = 0$ for $1 \leq i \leq n$, so as to avoid trivial solution $\boldsymbol{\alpha} = \mathbf{I}_n$ where $\mathbf{I}_n$ is a $n \times n$ identity matrix.

ℓ^1-graph constructs the sparse graph $G = (\boldsymbol{X}, \mathbf{W})$ where $\boldsymbol{X}$ is the set of vertices, $\mathbf{W}$ is the weighted adjacency matrix of G and $\mathbf{W}_{ij}$ indicates the edge

weight, or the similarity, between $\mathbf{x}_i$ and $\mathbf{x}_j$. $\mathbf{W}$ is set by the sparse codes:

$$\mathbf{W}_{ij} = (|\boldsymbol{\alpha}_{ij}| + |\boldsymbol{\alpha}_{ji}|)/2 \quad 1 \leq i, j \leq n \tag{3.4}$$

ℓ^1-graph then performs spectral clustering on the sparse similarity matrix $\mathbf{W}$ to obtain the data clusters. It should be emphasized that the above sparse graph construction method is used for almost all the sparse graph based clustering methods [2, 3, 46, 4, 49] while the sparse codes could be learned in different ways.

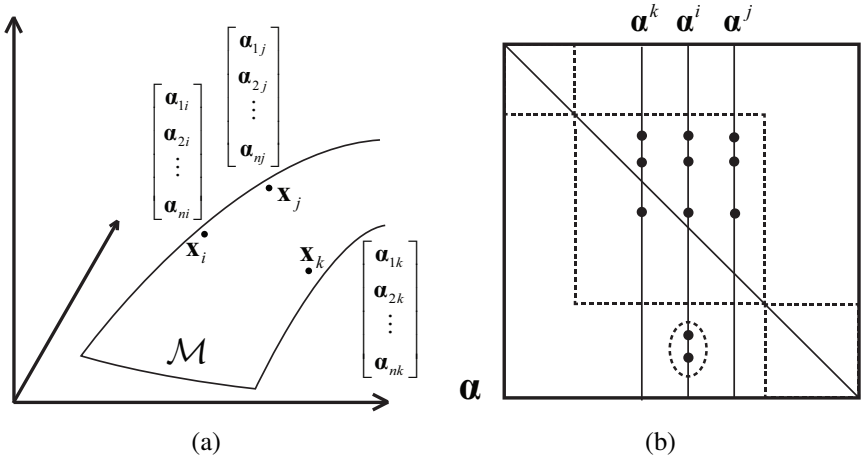


Figure 3.1: (a) Illustration of the manifold assumption used in our ℓ^0-ℓ^1-graph. This figure shows an example of a two-dimensional submanifold $\mathcal{M}$ in the three-dimensional ambient space. Three neighboring points $\mathbf{x}_i$, $\mathbf{x}_j$ and $\mathbf{x}_k$ in the submanifold are supposed to have similar sparse codes, i.e. $\boldsymbol{\alpha}^i = [\boldsymbol{\alpha}_{1i}, \ldots, \boldsymbol{\alpha}_{ni}]^\top$, $\boldsymbol{\alpha}^j = [\boldsymbol{\alpha}_{1j}, \ldots, \boldsymbol{\alpha}_{nj}]^\top$ and $\boldsymbol{\alpha}^k = [\boldsymbol{\alpha}_{1k}, \ldots, \boldsymbol{\alpha}_{nk}]^\top$, according to the manifold assumption. ℓ^0-distance $\|\boldsymbol{\alpha}^i - \boldsymbol{\alpha}^j\|_0$ is used to measure the distance between sparse codes for ℓ^0-ℓ^1-graph, while ℓ^2-distance $\|\boldsymbol{\alpha}^i - \boldsymbol{\alpha}^j\|_2$ is used for most existing sparse representation methods using graph regularization. (b) Illustration of the coefficient matrix $\boldsymbol{\alpha}$ comprising the sparse codes of all the data, where the black dots indicate nonzero elements, and the inner dashed box specifies the scope of correct neighbors, i.e. the ones in the same ground truth cluster. $\mathbf{x}_k$ and $\mathbf{x}_j$ choose the correct neighbors, and the local smoothness on the local sparse graph structure would encourage $\mathbf{x}_i$ to abandon the two wrong neighbors encompassed by the dashed ellipse.

High dimensional data always lie on or close to a submanifold of low intrinsic dimension, and clustering the data according to its underlying manifold structure is important and challenging in computer vision and machine learning. While ℓ^1-graph demonstrates better performance than many traditional similarity-based

clustering methods, it performs sparse representation for each datum independently without considering the geometric information and manifold structure of the entire data. On the other hand, in order to obtain the data embedding that accounts for the geometric information and manifold structure of the data, the manifold assumption [5] is usually employed [6, 7, 8, 9]. Interpreting the sparse code of a data point as its embedding, the manifold assumption in the case of sparse representation for most existing methods requires that if two points $\mathbf{x}_i$ and $\mathbf{x}_j$ are close in the intrinsic geometry of the submanifold, their corresponding sparse codes $\boldsymbol{\alpha}^i$ and $\boldsymbol{\alpha}^j$ are also expected to be similar to each other in the sense of ℓ^2-distance [8, 9]. In other words, $\boldsymbol{\alpha}$ varies smoothly along the geodesics in the intrinsic geometry (see Figure 3.1(a)). Based on the spectral graph theory [39], extensive literature uses graph Laplacian to impose local smoothness of the embedding and preserve the local manifold structure [5, 8, 9]. Given a proper symmetric similarity matrix $\mathbf{S}$, the sparse code $\boldsymbol{\alpha}$ that captures the local geometric structure of the data in accordance with the manifold assumption by graph Laplacian minimizes the following ℓ^2 regularization term:

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{S}_{ij}\|\boldsymbol{\alpha}^i-\boldsymbol{\alpha}^j\|_2^2=\mathrm{Tr}(\boldsymbol{\alpha}\mathbf{L}_{\mathbf{S}}\boldsymbol{\alpha}^\top) \tag{3.5}$$

where the ℓ^2-norm is used to measure the distance between sparse codes. $\mathbf{L}_{\mathbf{S}}=\mathbf{D}_{\mathbf{S}}-\mathbf{S}$ is the graph Laplacian using the adjacency matrix $\mathbf{S}$, the degree matrix $\mathbf{D}_{\mathbf{S}}$ is a diagonal matrix with each diagonal element being the sum of the elements in the corresponding row of $\mathbf{S}$, namely $(\mathbf{D}_{\mathbf{S}})_{ii}=\sum_{j=1}^{n}\mathbf{S}_{ij}$. To the best of our knowledge, such ℓ^2 regularization is employed by most methods that use graph regularization for sparse representation. *Incorporating the ℓ^2 regularization term into the optimization problem of ℓ^1-graph, the formulation of ℓ^2 graph regularization for ℓ^1-graph, which is also named ℓ^2-ℓ^1-graph*, is

$$\min_{\boldsymbol{\alpha}}L(\boldsymbol{\alpha})=\sum_{i=1}^{n}\|\mathbf{x}_i-\boldsymbol{X}\boldsymbol{\alpha}^i\|_2^2+\lambda_{\ell^1}\|\boldsymbol{\alpha}^i\|_1+\gamma_{\ell^2}\mathrm{Tr}(\boldsymbol{\alpha}\mathbf{L}_{\mathbf{S}}\boldsymbol{\alpha}^\top) \tag{3.6}$$

where $\gamma_{\ell^2}>0$ is the weighting parameter for the ℓ^2 regularization term. Following the representative ℓ^2 graph regularization method [8, 9], $\mathbf{S}$ is chosen as the adjacency matrix of K-nearest-neighbor (KNN) graph, i.e. $\mathbf{S}_{ij}=1$ if and only if $\mathbf{x}_i$ is among the K nearest neighbors of $\mathbf{x}_j$. Note that KNN is extensively

used in the manifold learning literature, such as Locally Linear Embedding [50], Laplacian Eigenmaps [51] and Sparse Manifold Clustering and Embedding [49], to establish the local neighborhood in the manifold. Although $\mathbf{S}$ is not symmetric, letting $\mathbf{S}' = \frac{\mathbf{S}+\mathbf{S}^\top}{2}$, then a symmetric adjacency matrix can be used in the graph regularization term without changing its value: $\mathrm{Tr}(\boldsymbol{\alpha}\mathbf{L}_{\mathbf{S}'}\boldsymbol{\alpha}^\top) = \mathrm{Tr}(\boldsymbol{\alpha}\mathbf{L}_{\mathbf{S}}\boldsymbol{\alpha}^\top)$.

In the following subsection, we propose ℓ^0-ℓ^1-graph, which uses ℓ^0-norm to measure the distance between the sparse codes in the graph regularization term in (3.6) based on the manifold assumption on the local structure of the sparse graph, leading to ℓ^0 graph regularization for ℓ^1-graph with superior clustering performance.

Algorithm 1 Data Clustering by ℓ^0-ℓ^1-Graph

Input:

The data set $\boldsymbol{X} = \{\mathbf{x}_i\}_{i=1}^n$, the number of clusters c, the parameter λ, γ, K for ℓ^0-ℓ^1-graph, λ_{ℓ^1} for the initialization of the ℓ^0-ℓ^1-graph, maximum iteration number M_c for coordinate descent, and maximum iteration number M_p for the iterative proximal method, stopping threshold ε

1: $r = 1$, initialize the coefficient matrix as $\boldsymbol{\alpha}^{(0)} = \boldsymbol{\alpha}_{\ell^1}$.
2: **while** $r \leq M_c$ **do**
3: Obtain $\boldsymbol{\alpha}^{(r)}$ from $\boldsymbol{\alpha}^{(r-1)}$ by coordinate descent. In i-th ($1 \leq i \leq n$) step of the r-th iteration of coordinate descent, solve (3.8) using the iterative proximal method (3.9) and (3.12) to update $\boldsymbol{\alpha}^i$ in each iteration of the proximal method.
4: **if** $|L(\boldsymbol{\alpha}^{(r)}) - L(\boldsymbol{\alpha}^{(r-1)})| < \varepsilon$ **then**
5: **break**
6: **else**
7: $r = r + 1$.
8: **end if**
9: **end while**
10: Obtain the sub-optimal coefficient matrix $\boldsymbol{\alpha}^*$ when the above iterations converge or maximum iteration number is achieved.
11: Build the pairwise similarity matrix by symmetrizing $\boldsymbol{\alpha}^*$: $\mathbf{W}^* = \frac{|\boldsymbol{\alpha}^*|+|\boldsymbol{\alpha}^*|^\top}{2}$, compute the corresponding normalized graph Laplacian $\mathbf{L}^* = (\mathbf{D}^*)^{-\frac{1}{2}}(\mathbf{D}^* - \mathbf{W}^*)(\mathbf{D}^*)^{-\frac{1}{2}}$, where $\mathbf{D}^*$ is a diagonal matrix with $\mathbf{D}^*_{ii} = \sum_{j=1}^{n} \mathbf{W}^*_{ij}$
12: Construct the matrix $\mathbf{v} = [\mathbf{v}_1, \ldots, \mathbf{v}_c] \in \mathbb{R}^{n\times c}$, where $\{\mathbf{v}_1, \ldots, \mathbf{v}_c\}$ are the c eigenvectors of $\mathbf{L}^*$ corresponding to its c smallest eigenvalues. Treat each row of $\mathbf{v}$ as a data point in $\mathbb{R}^c$, and run K-means clustering method to obtain the cluster labels for all the rows of $\mathbf{v}$.

Output: The cluster label of $\mathbf{x}_i$ is set as the cluster label of the i-th row of $\mathbf{v}$, $1 \leq i \leq n$.

3.3 The proposed ℓ^0-ℓ^1-Graph

Different from the previous graph regularized sparse representation methods [8, 9] where the sparse code of a data point serves as feature representation of that point for various learning tasks, the performance of sparse graph based clustering solely depends on the sparse graph. Since the only information of each data point in the sparse graph is its associated local structure of the sparse graph, rendering a sparse graph in accordance to the geometric information and manifold structure of the data requires the manifold assumption on the local sparse graph structure, mentioned in the introduction. This new variant of the manifold assumption encourages local smoothness on the local sparse graph structure. Such local smoothness is prone to produce the sparse graph complying to the manifold structure of the data. It also encourages the data points to coordinate with each other in selecting their neighbors. In the frequent case that most of the neighbors of a data point have nearly correct neighbor selection, the said local smoothness effectively advises this point to make a potentially better neighbor selection, compared to choosing neighbors on its own, especially when this point is subject to noise or itself is an outlier (see Figure 3.1(b)).

To facilitate optimization in terms of the sparse codes, we restrict the information of the local sparse graph structure for a data point to be that contained in its sparse code. Based on the construction of the sparse graph in Section 3.2, the local sparse graph structure contained in the sparse code of a data point is its support and nonzero elements: the support determines the neighbors it chooses and the nonzero elements contribute to the edge weights. Note that if the sparse codes of two data points have zero ℓ^0-distance, then they have similar local sparse graph structure. This motivates us to propose ℓ^0-ℓ^1-graph which employs ℓ^0-norm to measure the distance between sparse codes and promote local smoothness of the sparse graph structure. The optimization problem of ℓ^0-ℓ^1-graph is

$$\min_{\boldsymbol{\alpha}} L(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \|\mathbf{x}_i - \boldsymbol{X}\boldsymbol{\alpha}^i\|_2^2 + \lambda\|\boldsymbol{\alpha}^i\|_1 + \gamma \mathbf{R}_{\mathbf{S}}(\boldsymbol{\alpha}) \tag{3.7}$$

where $\mathbf{R}_{\mathbf{S}}(\boldsymbol{\alpha}) = \sum_{i,j=1}^{n} \mathbf{S}_{ij}\|\boldsymbol{\alpha}^i - \boldsymbol{\alpha}^j\|_0$ is the ℓ^0 regularization term, $\mathbf{S}$ is the adjacency matrix of the KNN graph, $\gamma > 0$ is the weighting parameter for ℓ^0 graph regularization term.

We use coordinate descent to optimize (3.7) with respect to $\boldsymbol{\alpha}^i$, i.e. in each step the i-th column of $\boldsymbol{\alpha}$, while fixing all the other sparse codes $\{\boldsymbol{\alpha}^j\}_{j\neq i}$. In each step

of coordinate descent, the optimization problem for $\boldsymbol{\alpha}^i$ is

$$\min_{\boldsymbol{\alpha}^i} F(\boldsymbol{\alpha}^i) = \|\mathbf{x}_i - \boldsymbol{X}\boldsymbol{\alpha}^i\|_2^2 + \lambda\|\boldsymbol{\alpha}^i\|_1 + \gamma \mathbf{R}_{\tilde{\mathbf{S}}}(\boldsymbol{\alpha}^i) \tag{3.8}$$

where $\mathbf{R}_{\tilde{\mathbf{S}}}(\boldsymbol{\alpha}^i) = \sum_{j=1}^{n} \tilde{\mathbf{S}}_{ij}\|\boldsymbol{\alpha}^i - \boldsymbol{\alpha}^j\|_0$, $\tilde{\mathbf{S}} = \mathbf{S} + \mathbf{S}^\top$

Inspired by recent advances in solving non-convex optimization problems by proximal linearized method [52], we propose an iterative proximal method to optimize the nonconvex problem (3.8). In the following text, the superscript with bracket indicates the iteration number of the proposed proximal method or the iteration number of the coordinate descent without confusion.

In t-th ($t \geq 1$) iteration of our proximal method, gradient descent is performed on the square loss term of (3.8), i.e. $P(\boldsymbol{\alpha}^i) = \|\mathbf{x}_i - \boldsymbol{X}\boldsymbol{\alpha}^i\|_2^2$:

$$\tilde{\boldsymbol{\alpha}^i}^{(t)} = \boldsymbol{\alpha}^{i(t-1)} - \frac{2}{\tau s}(\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\alpha}^{i(t-1)} - \boldsymbol{X}^\top \mathbf{x}_i) \tag{3.9}$$

where $\tau > 1$ is a constant and s is the Lipschitz constant for the gradient of function $P(\cdot)$, namely

$$\|\nabla P(\mathbf{Y}) - \nabla P(\mathbf{Z})\|_F \leq s\|\mathbf{Y} - \mathbf{Z}\|_F, \ \forall\, \mathbf{Y}, \mathbf{Z} \in \mathbb{R}^n \tag{3.10}$$

Then $\boldsymbol{\alpha}^{(t)}$ is obtained as the solution to the following ℓ^0-ℓ^1 regularized problem:

$$\boldsymbol{\alpha}^{i(t)} = \arg\min_{\mathbf{v}\in\mathbb{R}^n, \mathbf{v}_i = 0} \frac{\tau s}{2}\|\mathbf{v} - \tilde{\boldsymbol{\alpha}}^{(t)}\|_2^2 + \lambda\|\mathbf{v}\|_1 + \gamma \mathbf{R}_{\tilde{\mathbf{S}}}(\mathbf{v}) \tag{3.11}$$

Using the fact that $\max\{|\tilde{\boldsymbol{\alpha}}^{(t)}| - \frac{\lambda}{\tau s}, 0\} \circ \operatorname{sign}(\tilde{\boldsymbol{\alpha}}^{(t)})$ is the solution to

$$\arg\min_{\mathbf{v}} \frac{\tau s}{2}\|\mathbf{v} - \tilde{\boldsymbol{\alpha}}^{(t)}\|_2^2 + \lambda\|\mathbf{v}\|_1$$

where $\circ$ denotes element-wise multiplication, Proposition 1 below shows the closed form solution to the ℓ^0-ℓ^1 regularized subproblem (3.11):

Proposition 1. *Define* $F(\mathbf{v}_k) = \frac{\tau s}{2}\|\mathbf{v}_i - \tilde{\boldsymbol{\alpha}}_k^{(t)}\|_2^2 + \lambda|\mathbf{v}_k| + \gamma \mathbf{R}_{\tilde{\mathbf{S}}}(\mathbf{v}_k)$ *for* $\mathbf{v}_k \in \mathbb{R}$ *and* $\mathbf{R}_{\tilde{\mathbf{S}}}(\mathbf{v}_k) \triangleq \sum_{j=1}^{n} \tilde{\mathbf{S}}_{ij}\|\mathbf{v}_k - \boldsymbol{\alpha}_k^j\|_0$. *Let* $\mathbf{u} = \max\{|\tilde{\boldsymbol{\alpha}}^{(t)}| - \frac{\lambda}{\tau s}, 0\} \circ \operatorname{sign}(\tilde{\boldsymbol{\alpha}}^{(t)})$, *and let* $\mathbf{v}^*$ *be the optimal solution to (3.11). Then the* k*-th element of* $\mathbf{v}^*$ *is*

$$\mathbf{v}_k^* = \begin{cases} \arg\min_{\mathbf{v}_k \in \{\mathbf{u}_k\} \cup \{\boldsymbol{\alpha}_k^j\}_{\{j:\tilde{\mathbf{S}}_{ij}\neq 0\}}} F(\mathbf{v}_k) & : \quad k \neq i \\ 0 & : \quad k = i \end{cases} \tag{3.12}$$

Proposition 1 suggests an efficient way of obtaining the solution to (3.11). According to (3.12), $\boldsymbol{\alpha}^{i^{(t)}} = \mathbf{v}^*$ can be obtained by searching over a candidate set of size $K+1$, where K is the number of nearest neighbors to construct the KNN graph $\mathbf{S}$ for ℓ^0-ℓ^1-graph.

The iterative proximal method starts from $t = 1$ and continues until the sequence $\{F(\boldsymbol{\alpha}^{i^{(t)}})\}$ converges or maximum iteration number is achieved. When the proximal method converges or terminates for each $\boldsymbol{\alpha}^i$, the step of coordinate descent for $\boldsymbol{\alpha}^i$ is finished and the optimization algorithm proceeds to optimize other sparse codes. We initialize $\boldsymbol{\alpha}$ as $\boldsymbol{\alpha}^{(0)} = \boldsymbol{\alpha}_{\ell^1}$ and $\boldsymbol{\alpha}_{\ell^1}$ is the sparse codes generated by ℓ^1-graph by solving (3.3) with some proper weighting parameter λ_{ℓ^1}. In all the experimental results shown in the next section, we empirically set $\lambda_{\ell^1} = 0.1$.

The data clustering algorithm by ℓ^0-ℓ^1-graph is described in Algorithm 1. Letting the maximum iteration number for coordinate descent be M_c, and maximum iteration number M_p for each step of the coordinate descent, then the time complexity of running the coordinate descent for ℓ^0-ℓ^1-graph is $\mathcal{O}(M_c M_p n^3)$. Moreover, the following theorem shows that with a properly chosen s for gradient descent in (3.9), each iteration of the proposed proximal method decreases the value of the objective function $F(\cdot)$ in (3.8). Since each step of coordinate descent decreases the objective L, our coordinate descent method optimizing L always converges.

Theorem 3. *Let $s = 2\sigma_{\max}(\boldsymbol{X}^\top \boldsymbol{X})$ where $\sigma_{\max}(\cdot)$ indicates the largest eigenvalue of a matrix, then the sequence $\{F(\boldsymbol{\alpha}^{i^{(t)}})\}$ generated by the proximal method with (3.9) and (3.12) decreases, and the following inequality holds for $t \geq 1$:*

$$F(\boldsymbol{\alpha}^{i^{(t)}}) \leq F(\boldsymbol{\alpha}^{i^{(t-1)}}) - \frac{(\tau - 1)s}{2} \|\boldsymbol{\alpha}^{i^{(t)}} - \boldsymbol{\alpha}^{i^{(t-1)}}\|_F^2 \tag{3.13}$$

It follows that the sequence $\{F(\boldsymbol{\alpha}^{i^{(t)}})\}_t$ converges as a sequence indexed by t for each $1 \leq i \leq n$, so our proximal method converges.

Table 3.1: Clustering Results on Three UCI Data Sets

Data Set	Measure	KM	SC	ℓ^1-graph	SMCE	ℓ^2-ℓ^1-graph	ℓ^0-ℓ^1-graph
Heart	AC	0.5889	0.6037	0.6370	0.5963	0.6259	**0.6481**
	NMI	0.0182	0.0269	0.0529	0.0255	0.0475	**0.0637**
Ionosphere	AC	0.7095	0.7350	0.5071	0.6809	0.7236	**0.7635**
	NMI	0.1285	0.2155	0.1117	0.0871	0.1621	**0.2355**
Breast	AC	0.8541	0.8822	0.9033	0.8190	**0.9051**	**0.9051**
	NMI	0.4223	0.4810	0.5258	0.3995	0.5249	**0.5333**

Table 3.2: Clustering Results on COIL-20 Database

COIL-20 / # Clusters	Measure	KM	SC	ℓ^1-graph	SMCE	ℓ^2-ℓ^1-graph	ℓ^0-ℓ^1-graph
K = 4	AC	0.6625	0.6701	**1.0000**	0.7639	0.7188	**1.0000**
	NMI	0.5100	0.5455	**1.0000**	0.6741	0.6129	**1.0000**
K = 8	AC	0.5157	0.4514	0.7986	0.5365	0.6858	**0.9705**
	NMI	0.5342	0.4994	0.8950	0.6786	0.6927	**0.9581**
K = 12	AC	0.5823	0.4954	0.7697	0.6806	0.7512	**0.8333**
	NMI	0.6653	0.6096	0.8960	0.8066	0.7836	**0.9160**
K = 16	AC	0.6689	0.4401	0.8264	0.7622	0.8142	**0.8750**
	NMI	0.7552	0.6032	0.9294	0.8730	0.8511	**0.9435**
K = 20	AC	0.6504	0.4271	0.7854	0.7549	0.7771	**0.8208**
	NMI	0.7616	0.6202	0.9148	0.8754	0.8534	**0.9297**

Table 3.3: Clustering Results on COIL-100 Database

COIL-100 / # Clusters	Measure	KM	SC	ℓ^1-graph	SMCE	ℓ^2-ℓ^1-graph	ℓ^0-ℓ^1-graph
K = 20	AC	0.5875	0.4493	0.5340	0.6208	0.6681	**0.9250**
	NMI	0.7448	0.6680	0.7681	0.7993	0.7933	**0.9682**
K = 40	AC	0.5774	0.4160	0.5819	0.6028	0.5944	**0.8465**
	NMI	0.7662	0.6682	0.7911	0.7919	0.7991	**0.9484**
K = 60	AC	0.5330	0.3225	0.5824	0.5877	0.6009	**0.7968**
	NMI	0.7603	0.6254	0.8310	0.7971	0.8059	**0.9323**
K = 80	AC	0.5062	0.3135	0.5380	0.5740	0.5632	**0.7970**
	NMI	0.7458	0.6071	0.8034	0.7931	0.7934	**0.9240**
K = 100	AC	0.4928	0.2833	0.5310	0.5625	0.5493	**0.7425**
	NMI	0.7522	0.5913	0.8015	0.8057	0.8055	**0.9105**

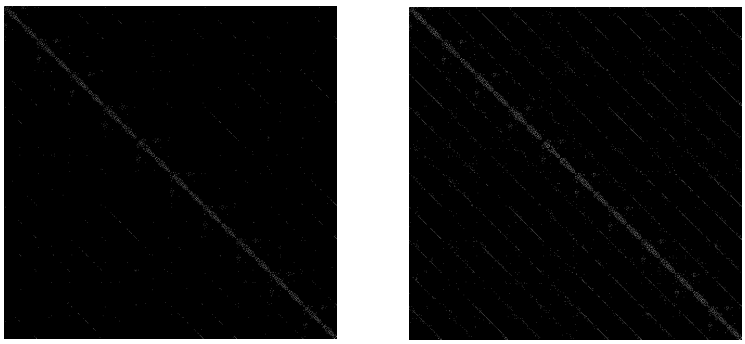


Figure 3.2: The comparison between the weighed adjacency matrix W of the sparse graph produced by ℓ^1-graph (right) and ℓ^0-ℓ^1-graph (left) on the Extended Yale Face Database B, where each white dot indicates an edge in the sparse graph.

3.4 Experimental Results

The superior clustering performance of ℓ^0-ℓ^1-graph is demonstrated by extensive experimental results on various data sets. ℓ^0-ℓ^1-graph is compared to K-means (KM), Spectral Clustering (SC), ℓ^1-graph, Sparse Manifold Clustering and Em-

Table 3.4: Clustering Results on the Extended Yale Face Database B.

Yale-B / # Clusters	Measure	KM	SC	ℓ^1-graph	SMCE	ℓ^2-ℓ^1-graph	ℓ^0-ℓ^1-graph
c = 10	AC	0.1780	0.1937	0.7580	0.3672	0.4563	**0.8750**
	NMI	0.0911	0.1278	0.7380	0.3264	0.4578	**0.8134**
c = 15	AC	0.1549	0.1748	0.7620	0.3761	0.4778	**0.7754**
	NMI	0.1066	0.1383	0.7590	0.3593	0.5069	**0.7814**
c = 20	AC	0.1227	0.1490	0.7930	0.3542	0.4635	**0.8376**
	NMI	0.0924	0.1223	0.7860	0.3789	0.5046	**0.8357**
c = 30	AC	0.1035	0.1225	0.8210	0.3601	0.5216	**0.8475**
	NMI	0.1105	0.1340	0.8030	0.3947	0.5628	**0.8652**
c = 38	AC	0.0948	0.1060	0.7850	0.3409	0.5091	**0.8500**
	NMI	0.1254	0.1524	0.7760	0.3909	0.5514	**0.8627**

Table 3.5: Clustering Results on CMU PIE Data

CMU PIE / # Clusters	Measure	KM	SC	ℓ^1-graph	SMCE	ℓ^2-ℓ^1-graph	ℓ^0-ℓ^1-graph
c = 20	AC	0.1327	0.1288	0.2329	0.2450	0.3076	**0.3294**
	NMI	0.1220	0.1342	0.2807	0.3047	0.3996	**0.4205**
c = 40	AC	0.1054	0.0867	0.2236	0.1931	0.3412	**0.3525**
	NMI	0.1534	0.1422	0.3354	0.3038	0.4789	**0.4814**
c = 68	AC	0.0829	0.0718	0.2262	0.1731	0.3012	**0.3156**
	NMI	0.1865	0.1760	0.3571	0.3301	**0.5121**	0.4800

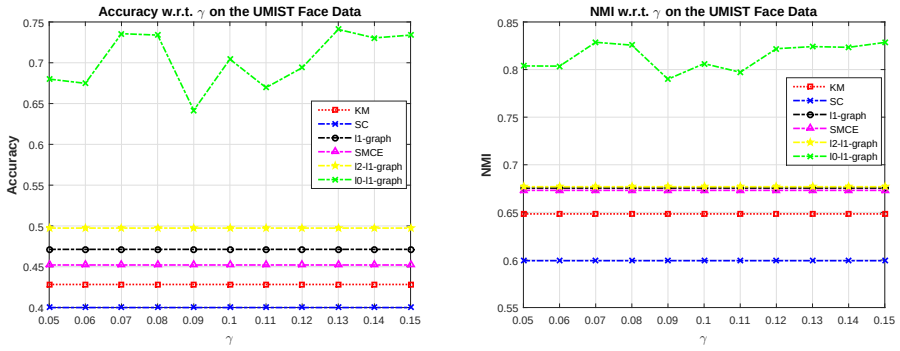


Figure 3.3: Clustering performance with different values of γ, i.e. the weight for the regularization term in ℓ^0-ℓ^1-graph, on the UMIST Face Data. Left: Accuracy; Right: NMI.

bedding (SMCE) [49], and ℓ^2-ℓ^1-graph introduced in Section 3.2.

3.4.1 Evaluation Metric

Two measures are used to evaluate the performance of the clustering methods: the accuracy and the Normalized Mutual Information (NMI) [53]. Let the predicted label of the datum $\mathbf{x}_i$ be $\hat{y}_i$ which is produced by the clustering method, and y_i is

Table 3.6: Clustering Results on CMU Multi-PIE which Contains the Facial Images Captured in Four Sessions (S1 to S4)

Data	Measure	KM	SC	ℓ^1-graph	SMCE	ℓ^2-ℓ^1-graph	ℓ^0-ℓ^1-graph
MPIE S1	AC	0.1167	0.1309	0.5892	0.1721	0.4173	**0.6815**
	NMI	0.5021	0.5289	0.7653	0.5514	0.7750	**0.8854**
MPIE S2	AC	0.1330	0.1437	0.6994	0.1898	0.5009	**0.7364**
	NMI	0.4847	0.5145	0.8149	0.5293	0.7917	**0.9048**
MPIE S3	AC	0.1322	0.1441	0.6316	0.1856	0.4853	**0.7138**
	NMI	0.4837	0.5150	0.7858	0.5155	0.7837	**0.8963**
MPIE S4	AC	0.1313	0.1469	0.6803	0.1823	0.5246	**0.7649**
	NMI	0.4876	0.5251	0.8063	0.5294	0.8056	**0.9220**

Table 3.7: Clustering Results on UMIST Face Data

UMIST Face / # Clusters	Measure	KM	SC	ℓ^1-graph	SMCE	ℓ^2-ℓ^1-graph	ℓ^0-ℓ^1-graph
K = 4	AC	0.4848	0.5691	0.4390	0.5203	**0.5854**	**0.5854**
	NMI	0.2889	0.4351	0.4645	0.3314	**0.4686**	0.4640
K = 8	AC	0.4330	0.4789	0.4836	0.4695	0.5399	**0.6948**
	NMI	0.5373	0.5236	0.5654	0.5744	0.5721	**0.7333**
K = 12	AC	0.4478	0.4655	0.4505	0.4955	0.5706	**0.6967**
	NMI	0.6121	0.6049	0.5860	0.6445	0.6994	**0.7929**
K = 16	AC	0.4297	0.4539	0.4124	0.4747	0.4700	**0.6544**
	NMI	0.6343	0.6453	0.6199	0.6909	0.6714	**0.7668**
K = 20	AC	0.4216	0.4174	0.4087	0.4452	0.4991	**0.7026**
	NMI	0.6377	0.6095	0.6111	0.6641	0.6893	**0.8038**

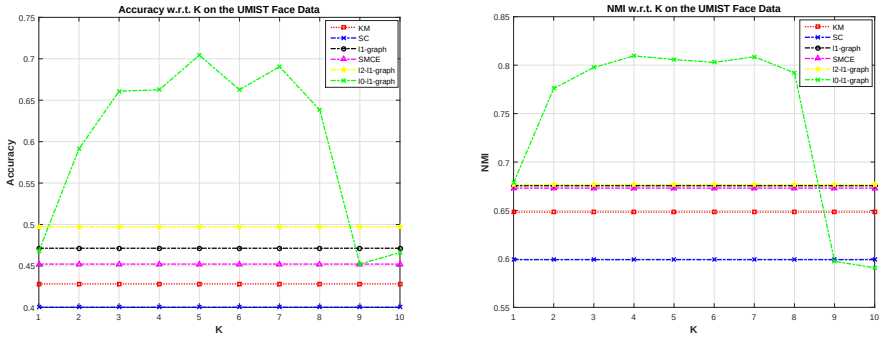


Figure 3.4: Clustering performance with different values of K, i.e. the number of nearest neighbors for the regularization term in ℓ^0-ℓ^1-graph, on the UMIST Face Data. Left: Accuracy; Right: NMI.

its ground truth label. The accuracy is defined as

$$Accuracy = \frac{\mathbb{1}_{\Omega(\hat{y}_i)\neq y_i}}{n} \tag{3.14}$$

where $\mathbb{1}$ is the indicator function, and Ω is the best permutation mapping function by the Kuhn-Munkres algorithm [54]. The more predicted labels match the ground truth ones, the more accuracy value is obtained.

Let $\hat{X}$ be the index set obtained from the predicted labels $\{\hat{y}_i\}_{i=1}^n$ and X be the index set from the ground truth labels $\{y_i\}_{i=1}^n$. The mutual information between $\hat{X}$ and X is

$$MI(\hat{X}, X) = \sum_{\hat{x}\in\hat{X}, x\in X} p(\hat{x}, x)\log_2(\frac{p(\hat{x}, x)}{p(\hat{x})p(x)}) \tag{3.15}$$

where $p(\hat{x})$ and $p(x)$ are the margined distribution of $\hat{X}$ and X respectively, induced from the joint distribution $p(\hat{x}, x)$ over $\hat{X}$ and X. Letting $H(\hat{X})$ and $H(X)$ be the entropy of $\hat{X}$ and X, then the normalized mutual information (NMI) is defined as

$$NMI(\hat{X}, X) = \frac{MI(\hat{X}, X)}{\max\{H(\hat{X}), H(X)\}} \tag{3.16}$$

It can be verified that the normalized mutual information takes values in $[0, 1]$. The accuracy and the normalized mutual information have been widely used for evaluating the performance of the clustering methods [8, 3, 53].

3.4.2 Clustering on UCI Data Sets

We conduct experiments on three real data sets from UCI machine learning repository [55], i.e. Heart, Ionosphere and Breast Cancer (Breast), to reveal the clustering performance of ℓ^0-ℓ^1-graph on general data sets. The clustering results on these three data sets are shown in Table 3.1.

3.4.3 Clustering On COIL-20 and COIL-100 Data

COIL-20 Database has 1440 images of resolution 32×32 for 20 objects, and the background is removed in all images. The dimension of this data is 1024. Its enlarged version, COIL-100 Database, contains 100 objects with 72 images of resolution 32×32 for each object. The images of each object were taken 5 degrees apart when each object was rotated on a turntable. The clustering results on these two data sets are shown in Tables 3.2 and 3.3 respectively. It can be observed that ℓ^2-ℓ^1-graph produces better clustering accuracy than ℓ^1-graph, since graph regularization produces locally smooth sparse codes aligned to the local manifold structure of the data. Using the ℓ^0-norm in the graph regularization term

to render the sparse graph that is better aligned to the geometric structure of the data, ℓ^0-ℓ^1-graph always performs better than all other competing methods.

3.4.4 Clustering on Yale-B, CMU PIE, CMU Multi-PIE, UMIST Face Data

The Extended Yale Face Database B contains face images for 38 subjects with 64 frontal face images taken under different illuminations for each subject. CMU PIE face data contains cropped face images of size 32×32 for 68 persons, and there are around 170 facial images for each person under different illumination and expressions, with a total number of 11554 images. CMU Multi-PIE (MPIE) data [56] contains the facial images captured in four sessions. The UMIST Face Database consists of 575 images of size 112×92 for 20 people. Each person is shown in a range of poses from profile to frontal views, each in a separate directory labelled $1a, 1b, \ldots, 1t$, and images are numbered consecutively as they were taken. The clustering results on these four face data sets are shown in Table 3.4, Table 3.5, Table 3.6 and Table 3.7 respectively. We conduct an extensive experiment on the popular face data sets in this subsection, and we observe that ℓ^0-ℓ^1-graph always achieves the highest accuracy, and best NMI for most cases, revealing the outstanding performance of our method and the effectiveness of manifold regularization on the local sparse graph structure. Figure 3.2 demonstrates that the sparse graph generated by ℓ^0-ℓ^1-graph effectively removes many incorrect neighbors for many data points through local smoothness of the sparse graph structure, compared to ℓ^1-graph.

3.4.5 Parameter Setting

There are two essential parameters for ℓ^0-ℓ^1-graph, i.e. γ for the ℓ^0 regularization term and K for building the adjacency matrix of the KNN graph. We use the sparse codes generated by ℓ^1-graph with weighting parameter $\lambda_{\ell^1} = 0.1$ in (3.3) to initialize both ℓ^0-ℓ^1-graph and ℓ^2-ℓ^1-graph, and set $\lambda = \gamma = 0.1$ in (3.7) and $K = 5$ for ℓ^0-ℓ^1-graph empirically throughout all the experiments. The maximum iteration number $M = 100$ and the stopping threshold $\varepsilon = 10^{-5}$.

In order to investigate how the performance of ℓ^0-ℓ^1-graph varies with parameter γ and K, we vary the weighting parameter γ and K, and illustrate the result

in Figures 3.3 and 3.4 respectively. The performance of ℓ^0-ℓ^1-graph is noticeably better than other competing algorithms over a relatively large range of both λ and K, which demonstrates the robustness of our algorithm with respect to the parameter settings. We also note that a too small K (near to 1) or too big K (near to 10) results in under regularization and over regularization.

3.4.6 Efficient Parallel Computing by CUDA Implementation

We have implemented ℓ^0-ℓ^1-graph in both MATLAB and CUDA C programming language on the cutting edge GPU, NVIDIA K40. Although the coordinate descent algorithm employed in Algorithm 1 cannot be parallelized since the result of each step of coordinate descent depends on other steps, we manage to implement all the matrix operations in the proposed proximal method by CUDA C. Throughout all the experiments, we consistently observe a speedup of around 20 times by our CUDA C implementation, compared to the well designed MATLAB implementation. Both the MATLAB and CUDA C code will be available for downloading.

3.5 Conclusion

We propose a novel ℓ^0-ℓ^1-graph for data clustering, which employs manifold assumption to align the sparse codes of ℓ^1-graph to the manifold structure of the original data. In contrast to most existing methods that use ℓ^2-norm to measure the distance between sparse codes in graph regularization for sparse representation, ℓ^0-ℓ^1-graph employs ℓ^0-norm to measure the distance between sparse codes so as to impose the local smoothness of the local sparse graph structure, leading to a sparse graph better aligned to the manifold structure of the data. We use coordinate descent to optimize the objective function of ℓ^0-ℓ^1-graph and propose an iterative proximal method to perform each step of the coordinate. The effectiveness of ℓ^0-ℓ^1-graph for data clustering is demonstrated by extensive experiment on various real data sets.

3.6 Proof of Theorem 3

Proof. First of all, when s is twice the maximum eigenvalue of $\boldsymbol{X}^\top \boldsymbol{X}$, then s is the Lipschitz constant for the gradient of function P. To see this, with $P(\mathbf{Y}) = \|\mathbf{x}_i - \boldsymbol{X}\mathbf{Y}\|_2^2$, we have $\nabla P(\mathbf{Y}) = 2(\boldsymbol{X}^\top \boldsymbol{X}\mathbf{Y} - \boldsymbol{X}^\top \mathbf{x}_i)$, and

$$\begin{aligned} &\|P(\mathbf{Y}) - \nabla P(\mathbf{Z})\|_2 = 2\|\boldsymbol{X}^\top \boldsymbol{X}(\mathbf{Y} - \mathbf{Z})\|_2 \\ &\leq 2\sigma_{\max}(\boldsymbol{X}^\top \boldsymbol{X}) \cdot \|(\mathbf{Y} - \mathbf{Z})\|_2 \end{aligned} \tag{3.17}$$

Define $f(\mathbf{v}) = \lambda\|\mathbf{v}\|_1 + \gamma \mathbf{R}_{\mathbf{S}}(\mathbf{v})$.
Since $\boldsymbol{\alpha}^{i(t)} = \underset{\mathbf{v}\in\mathbb{R}^n, \mathbf{v}_i = 0}{\arg\min} \frac{\gamma s}{2}\|\mathbf{v} - \tilde{\boldsymbol{\alpha}}^{i(t)}\|_2^2 + f(\mathbf{v})$,

$$\begin{aligned} &\frac{\tau s}{2}\|\boldsymbol{\alpha}^{i(t)} - \tilde{\boldsymbol{\alpha}}^{i(t)}\|_2^2 + f(\boldsymbol{\alpha}^{i(t)}) \\ &\leq \frac{\tau s}{2}\|\frac{\nabla P(\boldsymbol{\alpha}^{i(t-1)})}{\tau s}\|_2^2 + f(\boldsymbol{\alpha}^{i(t-1)}) \end{aligned} \tag{3.18}$$

which is equivalent to

$$\begin{aligned} &\langle \nabla P(\boldsymbol{\alpha}^{i(t-1)}), \boldsymbol{\alpha}^{i(t)} - \boldsymbol{\alpha}^{i(t-1)} \rangle + \frac{\tau s}{2}\|\boldsymbol{\alpha}^{i(t)} - \boldsymbol{\alpha}^{i(t-1)}\|_F^2 + f(\boldsymbol{\alpha}^{i(t)}) \\ &\leq f(\boldsymbol{\alpha}^{i(t-1)}) \end{aligned} \tag{3.19}$$

Also, since s is the Lipschitz constant for ∇P,

$$\begin{aligned} &P(\boldsymbol{\alpha}^{i(t)}) \leq P(\boldsymbol{\alpha}^{i(t-1)}) + \langle \nabla P(\boldsymbol{\alpha}^{i(t-1)}), \boldsymbol{\alpha}^{i(t)} - \boldsymbol{\alpha}^{i(t-1)} \rangle \\ &+ \frac{s}{2}\|\boldsymbol{\alpha}^{i(t)} - \boldsymbol{\alpha}^{i(t-1)}\|_2^2 \end{aligned} \tag{3.20}$$

Combining (3.19) and (3.20), we have

$$\begin{aligned} &F(\boldsymbol{\alpha}^{i(t)}) \leq F(\boldsymbol{\alpha}^{i(t-1)}) \\ &- \frac{(\tau - 1)s}{2}\|\boldsymbol{\alpha}^{i(t)} - \boldsymbol{\alpha}^{i(t-1)}\|_2^2 \end{aligned} \tag{3.21}$$

And (A.46) is verified. Since the sequence $\{F(\boldsymbol{\alpha}^{i(t)})\}_t$ is decreasing as sequence indexed by t with lower bound 0, it must converge. □

CHAPTER 4

SUBSPACE LEARNING WITH ℓ^0-GRAPH

4.1 Introduction

High-dimensional data often lie in a set of low-dimensional subspaces in many practical scenarios. Based on this observation, subspace clustering algorithms [57] aim to partition the data such that data belonging to the same subspace are identified as one cluster. Among various subspace clustering algorithms, the ones that employ sparsity prior, such as Sparse Subspace Clustering (SSC) [4], have been proven to be effective in separating the data in accordance with the subspaces that the data lie in under certain assumptions.

Sparse subspace clustering methods construct a sparse graph by sparse representation of the data, where the vertices represent the data. *Subspace-sparse representation ensures that vertices corresponding to different subspaces are disconnected in the sparse graph, and such a sparse graph is named a subspace-consistent sparse graph*. A subspace-consistent sparse graph produces compelling performance with standard graph based learning methods such as spectral clustering [10] and label propagation [11] for semi-supervised learning. [4] proves that when the subspaces are independent or disjoint, then subspace-sparse representations can be obtained by solving the canonical sparse coding problem using data as the dictionary under certain conditions on the rank, or singular value of the data matrix and the principle angle between the subspaces respectively. Under the independence assumption on the subspaces, low rank representation [58, 59] is also proposed to recover the subspace structures. Relaxing the assumptions on the subspaces to allow overlapping subspaces, the Greedy Subspace Clustering [60] and the Low-Rank Sparse Subspace Clustering [61] achieve subspace-sparse representation with high probability. However, their results rely on the semi-random model or full-random model which assumes that the data in each subspace are generated i.i.d. uniformly on the unit sphere in that subspace as well as certain additional

conditions on the size and dimensionality of the data. In addition, the geometric analysis in [62] also adopts the semi-random model and it handles overlapping subspaces. Noisy SSC proposed in [47] handles noisy data that lie in disjoint or overlapping subspaces.

To avoid the non-convex optimization problem incurred by ℓ^0-norm, most of the sparse subspace clustering or sparse graph based clustering methods use ℓ^1-norm [2, 3, 49, 4, 63] or ℓ^2-norm with thresholding [64] to impose the sparsity on the constructed similarity graph. In addition, ℓ^1-norm has been widely used as a convex relaxation of ℓ^0-norm for efficient sparse coding algorithms [65, 66, 67]. On the other hand, sparse representation methods such as [40] that directly optimize objective function involving ℓ^0-norm demonstrate compelling performance compared to its ℓ^1-norm counterpart. It remains an interesting question whether sparse subspace clustering equipped with ℓ^0-norm, which is the origination of the sparsity that counts the number of nonzero elements, has an advantage in obtaining the subspace-sparse representation. In this chapter, we propose ℓ^0-induced sparse subspace clustering which employs ℓ^0-norm to enforce the sparsity of representation, and we present a novel ℓ^0-graph for optimization. This chapter offers two major contributions:

Theoretical Results on ℓ^0-Induced Almost Surely Subspace-Sparse Representation We present the theory of the ℓ^0-induced sparse subspace clustering (ℓ^0-SSC), which shows that ℓ^0-SSC renders subspace-sparse representation, and consequently the subspace-consistent sparse graph, almost surely under minimum assumptions on the underlying subspaces the data lie in, i.e. subspaces are distinct. To the best of our knowledge, this is the mildest assumption on the subspaces compared to most existing sparse subspace clustering methods. Furthermore, our theory presented in Theorem 8 assumes that the data in each subspace are generated i.i.d. from arbitrary continuous distribution supported on that subspace, which is milder than the assumption of semi-random model in [60] and [61] that assume the data are i.i.d. uniformly distributed on the unit sphere in each subspace. Moreover, we prove that under the general conditions in Theorem 8, finding subspace representation cannot be computationally cheaper than solving the corresponding ℓ^0 problem. In fact, if there is an algorithm that obtains subspace representation for each data point, then it can be used to get the optimal solution to the ℓ^0 problem for ℓ^0-SSC by an additional step of polynomial

complexity.

Efficient Optimization With Theoretical Guarantee The optimization problem of ℓ^0-SSC is NP-hard and it is impractical to directly pursue the global optimal solution. Instead, we develop a novel algorithm named ℓ^0-graph which obtains a sub-optimal solution by a new efficient proximal method which converges to the critical point of the original objective, and ℓ^0-graph uses the sub-optimal solution to build a sparse similarity matrix for clustering. The bound for the distance between the sub-optimal solution and the global optimal solution under the assumption of the sparse eigenvalue on the data is given.

Note that SSC-OMP [68] adopts Orthogonal Matching Pursuit (OMP) [42] to choose neighbors for each datum in the sparse graph, which can be interpreted as approximately solving a ℓ^0 problem. We implement SSC-OMP and name it OMP-graph which solves the ℓ^0 problem of ℓ^0-SSC by OMP. However, SSC-OMP does not present the nice theoretical properties of the ℓ^0-SSC. In contrast, ℓ^0-graph obtains a sub-optimal solution to the objective of ℓ^0-SSC, and we give theory about the distance between the sub-optimal solution and the global optimal solution to the ℓ^0-SSC problem under the assumption of sparse eigenvalues on the data matrix. Moreover, extensive experimental results show the significant performance advantage of ℓ^0-graph over the OMP-graph.

The remaining parts of the chapter are organized as follows. The representative subspace clustering methods, SSC [4], are introduced in the next subsection. The theoretical property of ℓ^0-SSC, detailed formulation of ℓ^0-graph and theoretical guarantee on the obtained sub-optimal solution are illustrated. We then show the clustering and semi-supervised learning performance of the proposed ℓ^0-graph, and conclude the chapter. We use bold letters for matrices and vectors, and regular lowercase letters for scalars throughout this chapter. A bold letter with superscript indicates the corresponding column of a matrix, and a bold letter with subscript indicates the corresponding element of a matrix or vector. $\|\cdot\|_F$ and $\|\cdot\|_p$ denote the Frobenius norm and the ℓ^p-norm, and $\mathrm{diag}(\cdot)$ indicates the diagonal elements of a matrix.

4.1.1 Sparse Subspace Clustering and ℓ^1-Graph

SSC [4] and ℓ^1-graph [2, 3] employ the broadly used sparse representation [43, 44, 45, 63] of the data to construct the sparse graph. With the data $\boldsymbol{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ where n is the size of the data and d is the dimensionality, SSC and ℓ^1-graph solves the following sparse coding problem:

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \quad s.t.\ \boldsymbol{X} = \boldsymbol{X}\boldsymbol{\alpha},\ \mathrm{diag}(\boldsymbol{\alpha}) = \mathbf{0} \tag{4.1}$$

Both SSC and ℓ^1-graph construct a sparse graph $G = (\boldsymbol{X}, \mathbf{W})$ where the data $\boldsymbol{X}$ are represented as vertices, $\mathbf{W}$ of size $n \times n$ is the weighed adjacency matrix of the sparse graph G and $\mathbf{W}_{ij}$ indicates the edge weight, or similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. Note that there is an edge between $\mathbf{x}_i$ and $\mathbf{x}_j$ iff $\mathbf{W}_{ij} \neq 0$. $\mathbf{W}$ is set by the sparse codes $\boldsymbol{\alpha}$ as

$$\mathbf{W}_{ij} = (|\boldsymbol{\alpha}_{ij}| + |\boldsymbol{\alpha}_{ji}|)/2 \quad 1 \leq i, j \leq n \tag{4.2}$$

Furthermore, if the underlying subspaces that the data lie in are independent or disjoint, [4] proves that the optimal solution to (4.1) is the subspace-sparse representation under several additional conditions. *The sparse representation $\boldsymbol{\alpha}$ is called subspace-sparse representation if the nonzero elements of $\boldsymbol{\alpha}^i$, namely the sparse representation of the datum $\mathbf{x}_i$, correspond to the data points in the same subspace as $\mathbf{x}_i$.* Therefore, vertices corresponding to different subspaces are disconnected in the sparse graph. With the subsequent spectral clustering [10] applied on such sparse graph, compelling clustering performance is achieved.

Allowing some tolerance for inexact representation, robust sparse subspace clustering methods such as [47, 48] turn to solve the following Lasso-type problem for SSC and ℓ^1-graph:

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \quad s.t.\ \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{\alpha}\|_F \leq \delta,\ \mathrm{diag}(\boldsymbol{\alpha}) = \mathbf{0}$$

which is equivalent to the following problem

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{\alpha}\|_F^2 + \lambda_{\ell^1}\|\boldsymbol{\alpha}\|_1 \quad s.t.\ \mathrm{diag}(\boldsymbol{\alpha}) = \mathbf{0} \tag{4.3}$$

where $\lambda_{\ell^1} > 0$ is a weighting parameter for the ℓ^1 term.

Table 4.1: Assumptions on the subspaces and random data generation (for randomized part of the algorithm) for different subspace clustering methods. Note that $S_1 < S_2 < S_3 < S_4$, $D_1 < D_2$, where the assumption on the right hand side of $<$ is milder than that on the left hand side. The methods that are based on these assumptions are listed as follows. S_1: [58, 59]; S_2:[4]; S_3:[60, 61, 47, 62]; D_1: The data in each subspace are generated i.i.d. uniformly on the unit sphere in that subspace [60, 61, 62, 48]. D_2: The data in each subspace are generated i.i.d. from arbitrary continuous distribution supported on that subspace.

Assumption on Subspaces	Explanation
S_1:Independent Subspaces	$\mathrm{Dim}[\mathcal{S}_1 \otimes \mathcal{S}_2 \ldots \mathcal{S}_K] = \sum_k \mathrm{Dim}[\mathcal{S}_k]$
S_2:Disjoint Subspaces	$\mathcal{S}_k \cap \mathcal{S}_{k'} = \mathbf{0}$ for $k \neq k'$
S_3:Overlapping Subspaces	$1 \leq \mathrm{Dim}[\mathcal{S}_k \cap \mathcal{S}_{k'}] < \min\{\mathrm{Dim}[\mathcal{S}_k], \mathrm{Dim}[\mathcal{S}_{k'}]\}$ for $k \neq k'$
S_4:Distinct Subspaces (ℓ^0-Graph)	$\mathcal{S}_k \neq \mathcal{S}_{k'}$ for $k \neq k'$
Assumption on Random Data Generation	**Explanation**
D_1:Semi-Random Model or Full-Random Model	See caption above
D_2:IID (ℓ^0-Graph)	See caption above

4.2 ℓ^0-Induced Sparse Subspace Clustering

In this chapter, we investigate ℓ^0-induced sparse subspace clustering method, which solves the following ℓ^0 problem:

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \quad s.t.\ \boldsymbol{X} = \boldsymbol{X}\boldsymbol{\alpha},\ \mathrm{diag}(\boldsymbol{\alpha}) = \mathbf{0} \tag{4.4}$$

And the solution to the above problem is used to build a sparse graph for clustering as that for ℓ^1-graph. We then give the theorem about ℓ^0-induced almost surely subspace-sparse representation, and the proof is presented in the supplementary document for this chapter.

Theorem 4. *(ℓ^0-Induced Almost Surely Subspace-Sparse Representation) Suppose the data $\boldsymbol{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ lie in a union of K distinct subspaces $\{\mathcal{S}_k\}_{k=1}^K$ of dimensions $\{d_k\}_{k=1}^K$, i.e. $\mathcal{S}_k \neq \mathcal{S}_{k'}$ for $k \neq k'$. Let $\boldsymbol{X}^{(k)} \in \mathbb{R}^{d\times n_k}$ denote the data that belong to subspace $\mathcal{S}_k$, and $\sum_{k=1}^K n_k = n$. When $n_k \geq d_k + 1$, if the data belonging to each subspace are generated i.i.d. from arbitrary unknown continuous distribution supported on that subspace,*[1] *then with probability* 1*, the optimal solution to (A.39), denoted by $\boldsymbol{\alpha}^*$, is a subspace-sparse representation, i.e. nonzero elements in $\boldsymbol{\alpha}^{*i}$ corresponds to the data that lie in the same subspace as $\mathbf{x}_i$.*

[1] Continuous distribution here indicates that the data distribution is non-degenerate in the sense that the probability measure of any hyperplane of dimension less than that of the subspace is 0.

According to Theorem 8, ℓ^0-SSC (A.39) obtains the subspace-sparse representation almost surely under minimum assumption on the subspaces; i.e. it only requires that the subspaces be distinct. To the best of our knowledge, this is the mildest assumption on the subspaces for most existing sparse subspace clustering methods. Moreover, the only assumption on the data generation is that the data in each subspace are i.i.d. random samples from arbitrary continuous distributions supported on that subspace. In the light of assumed data distribution, such assumption on the data generation is much milder than the assumption of the semi-random model in ([60, 61, 62]) (note that the data can always be normalized to have unit norm and reside on the unit sphere). Table 4.1 summarizes different assumptions on the subspaces and random data generation for different subspace clustering methods including sparse subspace clustering methods. It can be seen that ℓ^0-SSC has mildest assumption on both subspaces and the random data generation.

The ℓ^0 sparse representation problem (A.39) is known to be NP-hard. One may ask if there is a shortcut to the almost surely subspace-sparse representation under the very mild assumption in Theorem 8. We show that such shortcut is almost surely impossible. Interestingly, the inverse of Theorem 8 also holds. Namely, suppose there is an algorithm which, for each data point $\mathbf{x}_i$, can find the data from the same subspace as $\mathbf{x}_i$ that linearly represent $\mathbf{x}_i$; then such representation almost surely leads to the solution to the corresponding ℓ^0 problem as

$$\min_{\boldsymbol{\alpha}^i} \|\boldsymbol{\alpha}^i\|_0 \quad s.t.\ \mathbf{x}_i = \boldsymbol{X}\boldsymbol{\alpha}^i,\ \boldsymbol{\alpha}_{ii} = 0 \tag{4.5}$$

Theorem 5. *(There is "no free lunch" for obtaining subspace representation under the general conditions of Theorem 8) Under the assumptions of Theorem 8, if there is an algorithm which, for any data point $\mathbf{x}_i \in \mathcal{S}_k$, $i \in [n], k \in [K]$, can find the data from the same subspace as $\mathbf{x}_i$ that linearly represent $\mathbf{x}_i$, i.e.*

$$\mathbf{x}_i = \boldsymbol{X}\boldsymbol{\beta} \quad (\boldsymbol{\beta}_i = 0) \tag{4.6}$$

where nonzero elements of $\boldsymbol{\beta}$ correspond to the data that lie in the subspace $\mathcal{S}_k$, then, with probability 1, the solution to the ℓ^0 problem (A.41) can be obtained from $\boldsymbol{\beta}$ in $\mathcal{O}(\hat{n}^3)$ time, where $\hat{n}$ is the number of nonzero elements in $\boldsymbol{\beta}$.

Therefore, we have the interesting "no free lunch" conclusion: With probability 1, finding the subspace representation for each data point cannot be much

computationally cheaper than solving the ℓ^0 sparse representation (A.39).

4.3 Optimization of ℓ^0-Graph

Similar to the case of SSC and ℓ^1-graph, by allowing tolerance for inexact representation, we turn to optimize the following ℓ^0 problem [2] for ℓ^0-SSC.

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^{n\times n},\mathrm{diag}(\boldsymbol{\alpha})=\mathbf{0}} L(\boldsymbol{\alpha}) = \|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{\alpha}\|_F^2 + \lambda\|\boldsymbol{\alpha}\|_0 \tag{4.7}$$

Problem (4.7) is NP-hard, and it is impractical to seek its global optimal solution. The literature extensively resorts to approximate algorithms, such as Orthogonal Matching Pursuit [42], or algorithms that use surrogate functions [69], for ℓ^0 problems. In this chapter we present ℓ^0-graph for data clustering, and ℓ^0-graph employs an iterative proximal method to optimize (4.7) and obtains a sub-optimal solution with theoretical guarantee. The sub-optimal solution is used to build a sparse similarity matrix for clustering in ℓ^0-graph. In the following text, the superscript with bracket indicates the iteration number of the proposed proximal method. Note that problem (4.7) is equivalent to a set of problems

$$\min_{\boldsymbol{\alpha}^i\in\mathbb{R}^n,\boldsymbol{\alpha}_i^i=0} L(\boldsymbol{\alpha}^i) = \|\mathbf{x}_i-\boldsymbol{X}\boldsymbol{\alpha}^i\|_2^2 + \lambda\|\boldsymbol{\alpha}^i\|_0 \tag{4.8}$$

for $1\leq i\leq n$. We describe the iterative proximal method for optimizing $L(\boldsymbol{\alpha}^i)$ with respect to the sparse code of the i-th data point, i.e. $\boldsymbol{\alpha}^i$, for any $1\leq i\leq n$. We initialize $\boldsymbol{\alpha}$ as $\boldsymbol{\alpha}^{(0)}=\boldsymbol{\alpha}_{\ell^1}$ and $\boldsymbol{\alpha}_{\ell^1}$ is the sparse codes generated by SSC or ℓ^1-graph via solving (4.3) with $\lambda_{\ell^1}=\lambda$. The data matrix $\boldsymbol{X}$ is normalized such that each column has unit ℓ^2-norm.

In t-th iteration of our proximal method for $t\geq 1$, gradient descent is performed on the squared loss term of $L(\boldsymbol{\alpha}^i)$, i.e. $Q(\boldsymbol{\alpha}^i)=\|\mathbf{x}_i-\boldsymbol{X}\boldsymbol{\alpha}^i\|_2^2$, to obtain

$$\tilde{\boldsymbol{\alpha}^i}^{(t)} = \boldsymbol{\alpha}^{i(t-1)} - \frac{2}{\tau s}(\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{\alpha}^{i(t-1)} - \boldsymbol{X}^\top\boldsymbol{X}) \tag{4.9}$$

where τ is any constant that is greater than 1. s is the Lipschitz constant for the

[2] Even if one sticks to the very original formulation without noise tolerance, (A.39) is still equivalent to (4.7) with some Lagrangian multiplier λ.

gradient of function $Q(\cdot)$, namely

$$\|\nabla Q(\mathbf{y}) - \nabla Q(\mathbf{z})\|_2 \leq s\|\mathbf{y} - \mathbf{z}\|_2,\ \forall\, \mathbf{y}, \mathbf{z} \in \mathbb{R}^n \tag{4.10}$$

s is usually chosen as two times the largest eigenvalue of $\boldsymbol{X}^\top \boldsymbol{X}$. Due to the sparsity of $\boldsymbol{\alpha}^i$, s can be much smaller which ensures the shrinkage of the support of the sequence $\{\boldsymbol{\alpha}^{i(t)}\}_t$ and the sparsity of the sub-optimal solution. $\boldsymbol{\alpha}^{i(t)}$ is then the solution to the following ℓ^0 regularized problem:

$$\begin{aligned} \boldsymbol{\alpha}^{i(t)} &= \arg\min_{\mathbf{v}\in\mathbb{R}^n} \frac{\tau s}{2}\|\mathbf{v} - \tilde{\boldsymbol{\alpha}}^{i(t)}\|_2^2 + \lambda\|\mathbf{v}\|_0 \\ s.t.\ \mathbf{v}_i &= \mathbf{0} \end{aligned} \tag{4.11}$$

It can be verified that (4.11) has closed-form solution, and the j-th element of $\boldsymbol{\alpha}^{i(t)}$ is

$$\boldsymbol{\alpha}_j^{i(t)} = \begin{cases} 0 & : \ |\tilde{\boldsymbol{\alpha}}_j^{i(t)}| < \sqrt{\frac{2\lambda}{\tau s}} \text{ or } i = j \\ \tilde{\boldsymbol{\alpha}}_j^{i(t)} & : \ \text{otherwise} \end{cases} \tag{4.12}$$

for $1 \leq j \leq n$. The iterations start from $t = 1$ and continue until the sequence $\{L(\boldsymbol{\alpha}^{i(t)})\}_t$ or $\{\boldsymbol{\alpha}^{i(t)}\}_t$ converges or maximum iteration number is achieved. With the obtained sub-optimal solution, a sparse similarity matrix is built as the weighted adjacency matrix of a sparse graph. Spectral clustering is performed upon such sparse graph to get the clustering result, as described in Algorithm 2 for ℓ^0-graph. The time complexity of our iterative proximal method is $\mathcal{O}(n^2 M_{\max})$ where $M_{\max}$ is the number of iterations (or maximum number of iterations) for the iterative proximal method.

4.4 Theoretical Analysis

In this section we present the bound for the distance between the sub-optimal solution by by ℓ^0-graph and the global optimal solution to the objective problem (A.52). We first prove that the sequence $\{\boldsymbol{\alpha}^{i(t)}\}_t$ produced by our iterative proximal method has shrinking support and the objective sequence $\{L(\boldsymbol{\alpha}^{i(t)})\}_t$ is decreasing so that it always converges in Lemma 7. Under the assumptions of sparse

eigenvalues on the data $\boldsymbol{X}$, we show that the sub-optimal solution by ℓ^0-graph is actually a critical point, namely $\{\boldsymbol{\alpha}^{i(t)}\}_t$ converges to a critical point of the objective (A.52); and this sub-optimal solution and the global optimal solution to (A.52) are local solutions of a carefully designed capped-ℓ^1 regularized problem. Based on established theory in [70] that deals with the distance between different local solutions, the bound for distance between the sub-optimal solution and the global optimal solution is presented in Theorem 10, again under the assumption of sparse eigenvalues on $\boldsymbol{X}$. Note that our analysis is valid for all $1 \leq i \leq n$.

In the following analysis, we let $\boldsymbol{\beta}_{\mathbf{I}}$ denote the vector formed by the elements of $\boldsymbol{\beta}$ with indices in $\mathbf{I}$ when $\boldsymbol{\beta}$ is a vector, or matrix formed by columns of $\boldsymbol{\beta}$ with indices in $\mathbf{I}$ when $\boldsymbol{\beta}$ is a matrix. Also, we let $\mathbf{S}_i = \text{supp}(\boldsymbol{\alpha}^{i(0)})$ and $|\mathbf{S}_i| = A_i$ for $1 \leq i \leq n$.

Lemma 4. *(Support Shrinkage in the Proximal Iterations and Sufficient Decrease of the Objective) When $s > \max\{2A_i, \frac{2(1+\lambda A_i)}{\lambda\tau}\}$, then the sequence $\{\boldsymbol{\alpha}^{i(t)}\}_t$ generated by the proximal method with (A.43) and (A.44) satisfies*

$$\text{supp}(\boldsymbol{\alpha}^{i(t)}) \subseteq \text{supp}(\boldsymbol{\alpha}^{i(t-1)}), t \geq 1 \tag{4.13}$$

namely the support of the sequence $\{\boldsymbol{\alpha}^{i(t)}\}_t$ shrinks when the iterative proximal proceeds. Moreover, the sequence of the objective $\{L(\boldsymbol{\alpha}^{i(t)})\}_t$ decreases, and the following inequality holds for $t \geq 1$:

$$L(\boldsymbol{\alpha}^{i(t)}) \leq L(\boldsymbol{\alpha}^{i(t-1)}) - \frac{(\tau-1)s}{2}\|\boldsymbol{\alpha}^{i(t)} - \boldsymbol{\alpha}^{i(t-1)}\|_2^2 \tag{4.14}$$

And it follows that the sequence $\{L(\boldsymbol{\alpha}^{i(t)})\}_t$ converges. The above results hold for any $1 \leq i \leq n$.

Before stating Lemma 8, the following definitions are introduced which are essential for our analysis.

Definition 2. *(Critical Points) Given the non-convex function $f\colon \mathbb{R}^n \to R \cup \{+\infty\}$ which is a proper and lower semi-continuous function:*

- *For a given $\mathbf{x} \in \text{dom} f$, its Frechet subdifferential of f at $\mathbf{x}$, denoted by $\tilde{\partial} f(x)$, is the set of all vectors $\mathbf{u} \in \mathbb{R}^n$ which satisfy*

$$\lim_{\mathbf{y}\neq\mathbf{x}} \inf_{\mathbf{y}\to\mathbf{x}} \frac{f(\mathbf{y}) - f(\mathbf{x}) - \langle \mathbf{u}, \mathbf{y}-\mathbf{x}\rangle}{\|\mathbf{y}-\mathbf{x}\|} \geq 0$$

- *The limiting-subdifferential of f at $\mathbf{x} \in \mathbb{R}^n$, denoted by written $\partial f(x)$, is defined by*

$$\begin{aligned}\partial f(x) = \{&\mathbf{u} \in \mathbb{R}^n : \exists \mathbf{x}^k \to \mathbf{x}, f(\mathbf{x}^k) \to f(\mathbf{x}),\\ &\tilde{\mathbf{u}}^k \in \tilde{\partial} f(\mathbf{x}_k) \to \mathbf{u}\}\end{aligned}$$

The point $\mathbf{x}$ is a critical point of f if $0 \in \partial f(x)$.

Also, we are considering the following capped-ℓ^1 regularized problem, which replaces the noncontinuous ℓ^0-norm with the continuous capped-ℓ^1 regularization term:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^n, \boldsymbol{\beta}_i = 0} L_{\text{capped}-\ell^1}(\boldsymbol{\beta}) = \|\mathbf{x}_i - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \mathbf{R}(\boldsymbol{\beta}; b) \tag{4.15}$$

where $\mathbf{R}(\boldsymbol{\beta}; b) = \sum_{j=1}^{n} R(\boldsymbol{\beta}_j; b)$, $R(t; b) = \lambda \frac{\min\{|t|, b\}}{b}$ for some $b > 0$. It can be seen that $R(t; b)$ approaches the ℓ^0 term when $b \to 0+$.

Now we define the local solution of the problem (4.15).

Definition 3. *(Local Solution) A vector $\tilde{\boldsymbol{\beta}}$ is a local solution to the problem (4.15) if*

$$\|\boldsymbol{X}^\top(\boldsymbol{X}^\top \tilde{\boldsymbol{\beta}} - \mathbf{x}_i) + \dot{\mathbf{R}}(\tilde{\boldsymbol{\beta}}; b)\|_2 = 0 \tag{4.16}$$

where $\dot{\mathbf{R}}(\tilde{\boldsymbol{\beta}}; b) = [\dot{R}(\tilde{\boldsymbol{\beta}}_1; b), \dot{R}(\tilde{\boldsymbol{\beta}}_2; b), \ldots, \dot{R}(\tilde{\boldsymbol{\beta}}_n; b)]^\top$.

Note that in the above definition and the following text, $\dot{R}(t; b)$ can be chosen as any value between the right differential $\frac{\partial R}{\partial t}(t+; b)$ (or $\dot{R}(t+; b)$) and left differential $\frac{\partial R}{\partial t}(t-; b)$ (or $\dot{R}(t-; b)$).

Definition 4. *(Sparse Eigenvalues) The lower and upper sparse eigenvalues of a matrix $\mathbf{A}$ is defined as*

$$\begin{aligned}\kappa_-(m) &:= \min_{\|\mathbf{u}\|_0 \le m; \|\mathbf{u}\|_2 = 1} \|\mathbf{A}\mathbf{u}\|_2^2\\ \kappa_+(m) &:= \max_{\|\mathbf{u}\|_0 \le m, \|\mathbf{u}\|_2 = 1} \|\mathbf{A}\mathbf{u}\|_2^2\end{aligned}$$

It is worthwhile mentioning that eigenvalues are closely related to the Restricted Isometry Property (RIP) [71] used frequently in the compressive sensing literature. Typical RIP requires bounds such as $\delta_\tau + \delta_{2\tau} + \delta_{3\tau} < 1$ or $\delta_{2\tau} < \sqrt{2} - 1$

[72] for stably recovering the signal from measurements and τ is the sparsity of the signal, where $\delta_\tau = \max\{\kappa_+(\tau) - 1, 1 - \kappa_-(\tau)\}$. Similar to [70], we use more general conditions on the sparse eigenvalues in this chapter (in the sense of not requiring bounds in terms of δ) to obtain theoretical results.

Definition 5. *(Degree of Nonconvexity of a Regularizer) For $\kappa \geq 0$ and $t \in \mathbb{R}$, define*

$$\theta(t, \kappa) := \sup_s\{-\text{sgn}(s-t)(\dot{P}(s; b) - \dot{P}(t; b)) - \kappa|s - t|\}$$

as the degree of nonconvexity for function P. If $\mathbf{u} = (u_1, \ldots, u_n)^\top \in \mathbb{R}^n$, $\theta(\mathbf{u}, \kappa) = [\theta(u_1, \kappa), \ldots, \theta(u_p, \kappa)]$.

Note that $\theta(t, \kappa) = 0$ for convex function P.

In the following lemma, we show that the sequence $\{\boldsymbol{\alpha}^{i(t)}\}_t$ generated by our proximal method converges to a critical point of $L(\boldsymbol{\alpha}^i)$, which is denoted by $\hat{\boldsymbol{\alpha}^i}$. And we denote by $\boldsymbol{\alpha}^{i*}$ the global optimal solution to (A.52), the ℓ^0-SSC problem for point $\mathbf{x}_i$. Let $\hat{\mathbf{S}}_i = \text{supp}(\hat{\boldsymbol{\alpha}^i})$, $\mathbf{S}_i^* = \text{supp}(\hat{\boldsymbol{\alpha}^i})$, then the following lemma also shows that both $\hat{\boldsymbol{\alpha}^i}$ and $\boldsymbol{\alpha}^{i*}$ are local solutions to the capped-ℓ^1 regularized problem (4.15).

Lemma 5. *(Solution by our proximal method and the global optimal solution to the ℓ^0 problem are local solutions of capped-ℓ^1 regularized problem) For any $1 \leq i \leq n$, suppose $\kappa_-(A_i) > 0$; then the sequence $\{\boldsymbol{\alpha}^{i(t)}\}_t$ generated by the proximal method with (A.43) and (A.44) converges to a critical point of $L(\boldsymbol{\alpha}^i)$, which is denoted by $\hat{\boldsymbol{\alpha}^i}$. Moreover, if*

$$0 < b < \min\{\min_{j \in \hat{\mathbf{S}}_i} |\hat{\boldsymbol{\alpha}_j^i}|, \frac{\lambda}{\max_{j \notin \hat{\mathbf{S}}_i} |\frac{\partial Q}{\partial \boldsymbol{\alpha}_j^i}|_{\boldsymbol{\alpha}^i = \hat{\boldsymbol{\alpha}^i}}|}, \tag{4.17}$$

$$\min_{j \in \mathbf{S}_i^*} |\boldsymbol{\alpha}_j^{i*}|, \frac{\lambda}{\max_{j \notin \mathbf{S}_i^*} |\frac{\partial Q}{\partial \boldsymbol{\alpha}_j^i}|_{\boldsymbol{\alpha}^i = \boldsymbol{\alpha}^{i*}}|}\} \tag{4.18}$$

(if the denominator is 0, $\frac{\lambda}{0}$ is defined to be $+\infty$ in this inequality), then both $\hat{\boldsymbol{\alpha}^i}$ and $\boldsymbol{\alpha}^{i*}$, i.e. the global optimal solution to (A.52), are local solutions to the capped-ℓ^1 regularized problem (4.15).

Theorem 5 in [70] gives the estimation on the distance between two local solutions of the capped-ℓ^1 regularized problem, based on which we have the following theorem showing that the sub-optimal solution $\hat{\boldsymbol{\alpha}^i}$ obtained by our proximal method is close to the global optimal solution to the original ℓ^0 problem (A.52), i.e. $\boldsymbol{\alpha}^{i*}$.

Theorem 6. *(Sub-optimal solution is close to the global optimal solution) For any* $1 \leq i \leq n$*, suppose* $\kappa_-(A_i) > 0$ *and* $\kappa_-(|\hat{\mathbf{S}}_i \cup \mathbf{S}_i^*|) > \kappa > 0$*, and* b *is chosen according to (5.24) as in Lemma 8. Then*

$$\|\boldsymbol{X}(\hat{\boldsymbol{\alpha}}^i - \boldsymbol{\alpha}^{i*})\|_2^2 \leq \frac{2\kappa_-(|\hat{\mathbf{S}}_i \cup \mathbf{S}_i^*|)}{(\kappa_-(|\hat{\mathbf{S}}_i \cup \mathbf{S}_i^*|) - \kappa)^2} \tag{4.19}$$
$$\big(\sum_{j \in \hat{\mathbf{S}}_i} (\max\{0, \frac{\lambda}{b} - \kappa|\hat{\boldsymbol{\alpha}}_j^i - b|\})^2 + |\mathbf{S}_i^* \setminus \hat{\mathbf{S}}_i|(\max\{0, \frac{\lambda}{b} - \kappa b\})^2\big)$$

In addition,

$$\|(\hat{\boldsymbol{\alpha}}^i - \boldsymbol{\alpha}^{i*})\|_2^2 \leq \frac{2}{(\kappa_-(|\hat{\mathbf{S}}_i \cup \mathbf{S}_i^*|) - \kappa)^2} \tag{4.20}$$
$$\big(\sum_{j \in \hat{\mathbf{S}}_i} (\max\{0, \frac{\lambda}{b} - \kappa|\hat{\boldsymbol{\alpha}}_j^i - b|\})^2 + |\mathbf{S}_i^* \setminus \hat{\mathbf{S}}_i|(\max\{0, \frac{\lambda}{b} - \kappa b\})^2\big)$$

Remark 1. *This result follows from Lemma 8 and Theorem 5 in [70]. The property of support shrinkage in Lemma 7 guarantees that* $\hat{\mathbf{S}}_i \subseteq \mathbf{S}_i$*, indicating that sub-optimal solution* $\hat{\boldsymbol{\alpha}}^i$ *is sparse, so we can expect that* $|\hat{\mathbf{S}}_i \cup \mathbf{S}_i^*|$ *is reasonably small. Also note that the bound for distance between the sub-optimal solution and the global optimal solution presented in Theorem 10 does not require typical RIP conditions. Also, when* $\frac{\lambda}{b} - \kappa|\hat{\boldsymbol{\alpha}}_j^i - b|$ *for nonzero* $\hat{\boldsymbol{\alpha}}_j^i$ *and* $\frac{\lambda}{b} - \kappa b$ *are no greater than* 0*, or they are small positive numbers, the sub-optimal solution* $\hat{\boldsymbol{\alpha}}^i$ *is equal to or very close to the global optimal solution.*

The detailed proofs of the theorems and lemmas in this chapter are included in the supplementary document (Section A.1).

Table 4.2: Clustering Results on UCI Ionosphere and Heart

Data Set	Measure	KM	SC	SSC	SMCE	SSC-OMP	Aℓ^0-SSC
Ionosphere	AC	0.7097	0.7350	0.5128	0.6809	0.6353	**0.7692**
	NMI	0.1287	0.2155	0.1165	0.0871	0.0299	**0.2609**
Heart	AC	0.5889	0.6037	0.6370	0.5963	0.5519	**0.6444**
	NMI	0.0182	0.0269	0.0529	0.0255	0.0058	**0.0590**

4.5 Experimental Results

The superior clustering and semi-supervised learning performance of ℓ^0-graph is demonstrated in this section. We compare our ℓ^0-graph to K-means (KM), Spectral Clustering (SC), SSC, Sparse Manifold Clustering and Embedding (SMCE)

Table 4.3: Clustering results on COIL-20 Database. c in the left column is the cluster number, i.e. the first c clusters of the entire data are used for clustering. c has the same meaning in Table 4.4 and Table 4.5.

COIL-20 / # Clusters	Measure	KM	SC	SSC	SMCE	OMP-Graph	ℓ^0-Graph
c = 4	AC	0.6632	0.6701	1.0000	0.7639	0.9271	**1.0000**
	NMI	0.5106	0.5455	1.0000	0.6741	0.8397	**1.0000**
c = 8	AC	0.5130	0.4462	0.7986	0.5365	0.6753	**0.9705**
	NMI	0.5354	0.4947	0.8950	0.6786	0.7656	**0.9638**
c = 12	AC	0.5885	0.4965	0.7697	0.6806	0.5475	**0.8310**
	NMI	0.6707	0.6096	0.8960	0.8066	0.6316	**0.9149**
c = 16	AC	0.6579	0.4271	0.8273	0.7622	0.3481	**0.9002**
	NMI	0.7555	0.6031	0.9301	0.8730	0.4520	**0.9552**
c = 20	AC	0.6554	0.4278	0.7854	0.7549	0.3389	**0.8472**
	NMI	0.7630	0.6217	0.9148	0.8754	0.4853	**0.9428**

Table 4.4: Clustering Results on COIL-100 Database.

COIL-100 / # Clusters	Measure	KM	SC	SSC	SMCE	OMP-Graph	ℓ^0-Graph
c = 20	AC	0.5850	0.4514	0.5757	0.6208	0.4243	**0.9264**
	NMI	0.7456	0.6700	0.7980	0.7993	0.5258	**0.9681**
c = 40	AC	0.5791	0.4139	0.5934	0.6038	0.2340	**0.8472**
	NMI	0.7691	0.6681	0.7962	0.7918	0.4378	**0.9471**
c = 60	AC	0.5371	0.3389	0.5657	0.5887	0.1905	**0.8326**
	NMI	0.7622	0.6343	0.8162	0.7973	0.3690	**0.9352**
c = 80	AC	0.5048	0.3115	0.5271	0.5835	0.2247	**0.7899**
	NMI	0.7474	0.6088	0.8006	0.8006	0.4173	**0.9218**
c = 100	AC	0.4996	0.2835	0.5275	0.5639	0.1667	**0.7683**
	NMI	0.7539	0.5923	0.8041	0.8064	0.3757	**0.9182**

[49]. Moreover, we derive the OMP-graph, which builds the sparse graph in the same way as ℓ^0-graph except that it solves the following optimization problem by Orthogonal Matching Pursuit (OMP) to obtain the sparse code for $1 \leq i \leq n$:

$$\min_{\boldsymbol{\alpha}^i} \|\mathbf{x}_i - \boldsymbol{X}\boldsymbol{\alpha}^i\|_F^2 \quad s.t.\ \|\boldsymbol{\alpha}^i\|_0 \leq T, \boldsymbol{\alpha}_i^i = 0 \tag{4.21}$$

ℓ^0-graph is also compared to OMP-graph to show the advantage of the proposed proximal method in the previous sections. By adjusting the parameters, ℓ^1-graph and SSC solve the same problem and generate equivalent results, so we report their performance under the same name "SSC". Two measures are used to evaluate the performance of the clustering methods, i.e. the accuracy and the Normalized Mutual Information (NMI) [53].

4.5.1 Clustering on UCI Data

In this subsection, we conduct experiments on the Ionosphere and Heart data from UCI machine learning repository [55], revealing the performance of Aℓ^0-SSC on

Table 4.5: Clustering Results on the Extended Yale Face Database B.

Yale-B / # Clusters	Measure	KM	SC	SSC	SMCE	OMP-Graph	ℓ^0-Graph
c = 10	AC	0.1782	0.1922	0.7580	0.3672	0.7375	**0.8406**
	NMI	0.0897	0.1310	0.7380	0.3266	0.7468	**0.7695**
c = 15	AC	0.1554	0.1706	0.7620	0.3761	0.7532	**0.7987**
	NMI	0.1083	0.1390	0.7590	0.3593	0.7943	**0.8183**
c = 20	AC	0.1200	0.1466	0.7930	0.3526	0.7813	**0.8273**
	NMI	0.0872	0.1183	0.7860	0.3771	0.8172	**0.8429**
c = 30	AC	0.1096	0.1209	0.8210	0.3470	0.7156	**0.8633**
	NMI	0.1159	0.1338	0.8030	0.3927	0.7260	**0.8762**
c = 38	AC	0.0954	0.1077	0.7850	0.3293	0.6529	**0.8480**
	NMI	0.1258	0.1485	0.7760	0.3812	0.7024	**0.8612**

Table 4.6: Clustering results on UMIST Face, CMU PIE, AR Face, CMU Multi-PIE and Georgia Tech Face database. Note that the CMU Multi-PIE contains the facial images captured in four sessions (S1 to S4).

Data	Measure	KM	SC	SSC	SMCE	OMP-Graph	ℓ^0-Graph
UMIST Face	AC	0.4275	0.4052	0.4904	0.4487	0.4835	**0.6730**
	NMI	0.6426	0.6159	0.6885	0.6696	0.6310	**0.7924**
CMU PIE	AC	0.0845	0.0729	0.2287	0.1733	0.0821	**0.2591**
	NMI	0.1884	0.1789	0.3659	0.3343	0.1494	**0.4435**
AR Face	AC	0.2752	0.2957	0.5914	0.3543	0.4229	**0.6086**
	NMI	0.5941	0.6248	0.8060	0.6573	0.6835	**0.8117**
MPIE S1	AC	0.1164	0.1285	0.5892	0.1721	0.1695	**0.6741**
	NMI	0.5049	0.5292	0.7653	0.5514	0.3395	**0.8622**
MPIE S2	AC	0.1315	0.1410	0.6994	0.1898	0.2093	**0.7527**
	NMI	0.4834	0.5128	0.8149	0.5293	0.4292	**0.8939**
MPIE S3	AC	0.1291	0.1459	0.6316	0.1856	0.1787	**0.7050**
	NMI	0.4811	0.5185	0.7858	0.5155	0.3415	**0.8750**
MPIE S4	AC	0.1308	0.1463	0.6803	0.1823	0.1680	**0.7246**
	NMI	0.4866	0.5280	0.8063	0.5294	0.3345	**0.8837**
Georgia Face	AC	0.4987	0.5187	0.5413	0.6053	0.4733	**0.6187**
	NMI	0.6856	0.7014	0.6968	0.7394	0.6622	**0.7400**

general machine learning data. The Ionosphere data contains 351 points of dimensionality 34. The Heart data contains 270 points of dimensionality 13. The clustering results on the two data sets are shown in Table 4.2.

4.5.2 Clustering On COIL-20 and COIL-100 Database

COIL-20 Database has 1440 images of 20 objects in which the background has been removed, and the size of each image is 32×32, so the dimension of this data is 1024. COIL-100 Database contains 100 objects with 72 images of size 32×32 for each object. The images of each object were taken 5 degrees apart when the object was rotated on a turntable. The clustering results on these two data sets are shown in Tables 4.3 and 4.4 respectively. We observe that ℓ^0-graph performs consistently better than all other competing methods. On COIL-100 Database, SMCE renders slightly better results than SSC on the entire data due to

Algorithm 2 Data Clustering by ℓ^0-Graph

Input:

The data set $\boldsymbol{X} = \{\mathbf{x}_i\}_{i=1}^n$, the number of clusters c, the parameter λ for ℓ^0-graph, maximum iteration number M, stopping threshold ε.

1: Initialize the coefficient matrix as $\boldsymbol{\alpha}^{(0)} = \boldsymbol{\alpha}_{\ell^1}$.
2: **for** $1 \leq i \leq n$ **do**
3: Obtain the sub-optimal solution $\hat{\boldsymbol{\alpha}^i}$ by the iterative proximal method with (A.43) and (A.44) starting from $t = 1$. The iteration terminates either $\{\boldsymbol{\alpha}^{i(t)}\}_t$ or $\{L(\boldsymbol{\alpha}^{i(t)})\}_t$ converges under the threshold ε or maximum iteration number is achieved (note that the optimization for $1 \leq i \leq n$ is performed in parallel).
4: **end for**
5: Obtain the sub-optimal coefficient matrix $\hat{\boldsymbol{\alpha}}$ where the $i-th$ column is $\hat{\boldsymbol{\alpha}^i}$ when the above iterations converge or maximum iteration number is achieved.
6: Build the sparse similarity matrix by symmetrizing $\hat{\boldsymbol{\alpha}}$: $\hat{\mathbf{W}} = \frac{|\hat{\boldsymbol{\alpha}}|+|\hat{\boldsymbol{\alpha}}^\top|}{2}$, compute the corresponding normalized graph Laplacian $\hat{\mathbf{L}} = (\hat{\mathbf{D}})^{-\frac{1}{2}}(\hat{\mathbf{D}} - \hat{\mathbf{W}})(\hat{\mathbf{D}})^{-\frac{1}{2}}$, where $\hat{\mathbf{D}}$ is a diagonal matrix with $\hat{\mathbf{D}}_{ii} = \sum_{j=1}^{n} \hat{\mathbf{W}}_{ij}$
7: Construct the matrix $\mathbf{v} = [\mathbf{v}_1, \ldots, \mathbf{v}_c] \in \mathbb{R}^{n \times c}$, where $\{\mathbf{v}_1, \ldots, \mathbf{v}_c\}$ are the c eigenvectors of $\mathbf{L}^*$ corresponding to its c smallest eigenvalues. Treat each row of $\mathbf{v}$ as a data point in $\mathbb{R}^c$, and run K-means clustering method to obtain the cluster labels for all the rows of $\mathbf{v}$.

Output: The cluster label of $\mathbf{x}_i$ is set as the cluster label of the i-th row of $\mathbf{v}$, $1 \leq i \leq n$.

its capability of modeling non-linear manifolds.

4.5.3 Clustering On Extended Yale Face Database B

The Extended Yale Face Database B contains face images for 38 subjects with 64 frontal face images taken under different illuminations for each subject. The clustering results are shown in Table 4.5. We can see that ℓ^0-graph achieves significantly better clustering result than SSC, which is the second best method on this data.

4.5.4 Clustering On More Face Data Sets

We demonstrate more experimental results on UMIST Face, CMU PIE, AR Face, CMU Multi-PIE and Georgia Tech Face Database in Table 4.6. The introduction

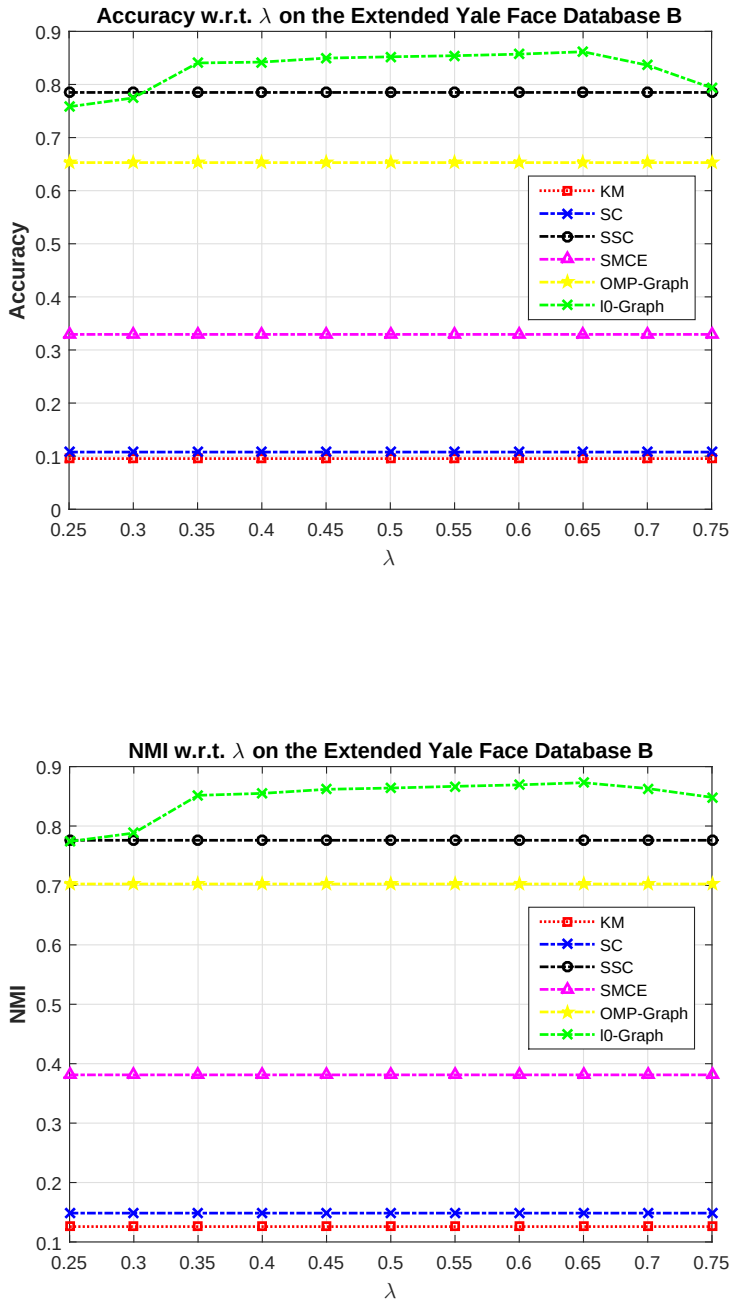


Figure 4.1: Clustering performance with different values of λ, i.e. the weight for the ℓ^0-norm, on the Extended Yale Face Database B. Left: Accuracy; Right: NMI.

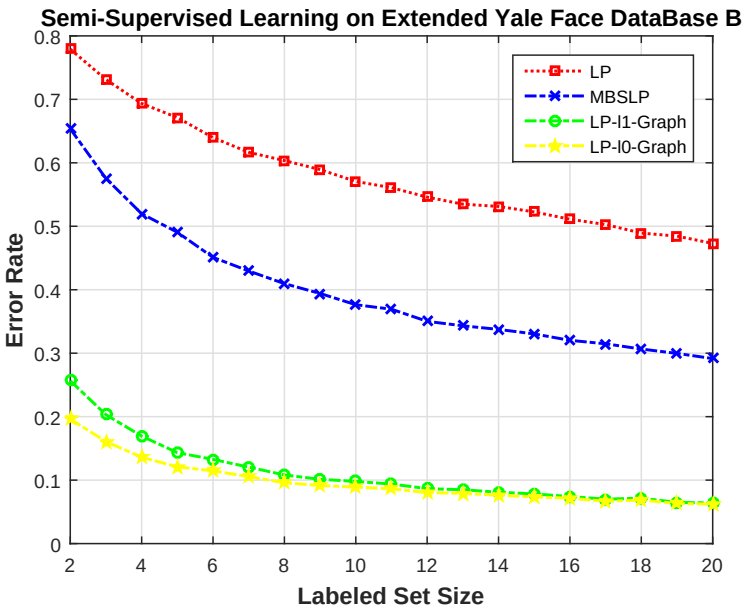


Figure 4.2: Semi-Supervised Learning by Label Propagation using Gaussian kernel graph (LP) [11], Manifold-based Similarity for Label Propagation (MBSLP) [73], Label Propagation using ℓ^1-graph (LP-ℓ^1-Graph), Label Propagation using ℓ^0-graph (LP-ℓ^0-Graph). The horizontal axis indicates the number of labeled samples for each class, and the vertical axis indicates the error rate of different methods.

to the data used in this table can be found at `http://www.face-rec.org/databases/`.

4.5.5 Parameter Setting

We set $\lambda = 0.5$ for ℓ^0-graph empirically throughout all the experiments in this section. The parameter T for OMP-graph in (4.21) is tuned to control the sparsity of the generated sparse codes such that the aforementioned average number of non-zero elements of the sparse code matches that of ℓ^0-graph. For SSC, the weighting parameter for the ℓ^1-norm is chosen from $[0.1, 1]$ for the best performance.

We investigate how the clustering performance on the Extended Yale Face Database B changes by varying the weighting parameter λ for ℓ^0-graph, and illustrate the result in Figure 4.1. The parameter sensitivity result on the COIL-20 Database is presented in the supplementary document (Section A.1). We observe that the performance of ℓ^0-graph is much better than other algorithms over a rel-

atively large range of λ, revealing the robustness of our algorithm with respect to the weighting parameter λ.

4.5.6 Semi-Supervised Learning Using ℓ^0-Graph

We demonstrate the performance of semi-supervised learning using SSC and ℓ^0-graph via Label Propagation [11], a widely used semi-supervised learning method which predicts the labels for unlabeled data by encouraging local smoothness of the labels in accordance with the similarity graph over the data. The performance of label propagation depends on the similarity graph. The comparison results for different semi-supervised learning methods on Extended Yale Face Database B are shown in Figure 4.2, which evidences the effectiveness of ℓ^0-graph for semi-supervised learning.

4.6 Conclusion

We propose a novel ℓ^0-graph for data clustering in this chapter under the principle of ℓ^0-induced sparse subspace clustering (ℓ^0-SSC). In contrast to the existing sparse subspace clustering method such as Sparse Subspace Clustering, ℓ^0-SSC features ℓ^0-induced almost surely subspace-sparse representation under milder assumptions on the subspaces and random data generation. The objective function of ℓ^0-SSC is optimized using a proposed proximal method in our novel ℓ^0-graph algorithm with theoretical guarantee. Extensive experimental results on various real data sets demonstrate the effectiveness and superiority of ℓ^0-graph over other competing methods.

CHAPTER 5

SUPPORT REGULARIZED SPARSE CODING AND ITS FAST ENCODER

5.1 Introduction

The aim of sparse coding is to represent an input vector by a linear combination of a few atoms of a learned dictionary which is usually over-complete, and the coefficients for the atoms are called sparse code. Sparse coding is widely applied in machine learning and signal processing, and sparse code is extensively used as a discriminative and robust feature representation with convincing performance for classification and clustering [43, 44, 45]. Suppose the data $\mathbf{D} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ lie in the d-dimensional Euclidean space $\mathbb{R}^d$, and the dictionary matrix is $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_p] \in \mathbb{R}^{d\times p}$ and each $\mathbf{d}_k\,(k = 1, \ldots, p)$ is an atom of the dictionary. Sparse coding seeks the linear sparse representation with respect to the dictionary $\mathbf{D}$ for each vector $\mathbf{x} \in \mathbf{D}$ by solving the following convex optimization problem:

$$\begin{aligned} &\min_{\mathbf{D},\mathbf{Z}} \sum_{i=1}^{n} \frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\mathbf{Z}^i\|_2^2 + \lambda\|\mathbf{Z}^i\|_1 \\ &s.t. \|\mathbf{D}^k\|_2 \le c_0, k = 1, \ldots, p \end{aligned} \tag{5.1}$$

where λ is a weighting parameter for the ℓ^1-norm of $\mathbf{z}$, and c_0 is a positive constraint that bounds the ℓ^2-norm of each dictionary atom. In [74], a feed-forward

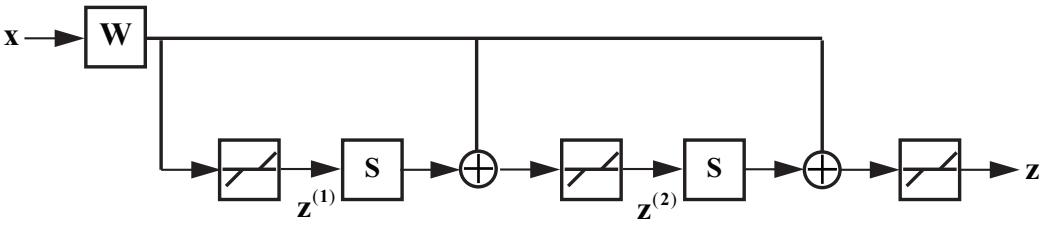


Figure 5.1: Illustration of LISTA network for approximate sparse coding.

neural network named Learned Iterative Shrinkage and Thresholding Algorithm (LISTA) is proposed to produce the approximation for sparse coding (5.1). The architecture of LISTA is illustrated in Figure 5.1. The LISTA network involves an finite number of stages wherein each stage performs the following operation on the intermediate sparse code:

$$\mathbf{z}^{(k+1)} = h_\theta(\mathbf{W}\mathbf{x} + \mathbf{S}\mathbf{z}^{(k)}), \quad \mathbf{z}^{(0)} = \mathbf{0} \tag{5.2}$$

where h_θ is an element-wise shrinkage function defined as

$$[h_\theta(\mathbf{u})]_k = \text{sign}(\mathbf{u}_k)(|\mathbf{u}_k| - \theta)_+, k = 1, \ldots, p \tag{5.3}$$

Let f indicate the LISTA network and it generates the approximate sparse code $\mathbf{z} = f(\mathbf{x}, \Theta)$, where $\Theta = (\mathbf{W}, \mathbf{S}, \theta)$ collectively denotes the parameters of the LISTA network. Supposing the optimal sparse codes for the training data $\mathbf{x}_1, \ldots, \mathbf{x}_m$ are $\mathbf{Z}^{*1}, \ldots, \mathbf{Z}^{*m}$, then the parameters Θ are learned by minimizing the following cost function which measures the distance between the predicted approximate sparse codes and the optimal sparse codes: $\mathcal{L}(\Theta) = \frac{1}{m}\sum_{i=1}^{m} \|\mathbf{Z}^{*i} - f(\mathbf{x}_i, \Theta)\|_2^2$. And the optimization is performed by stochastic gradient descent and back-propagation.

Sparse coding is widely used to model high-dimensional data. Based on the formulation of sparse coding (5.1), it can be observed that the sparse code for each data point is obtained independently when the dictionary is fixed, which ignores the geometric information and manifold structure of the high-dimensional data. In order to obtain the sparse code that accounts for the geometric information and manifold structure of the data, many regularized sparse coding methods, such as [6, 7, 8, 9], employ manifold assumption [5]. Manifold assumption in these methods imposes local smoothness on the sparse codes for nearby data; namely, nearby data are encouraged to have similar sparse codes in the sense of ℓ^2-distance, and they are termed ℓ^2-Regularized Sparse Coding (ℓ^2-RSC). In this paper, we propose Support Regularized Sparse Coding (SRSC). Compared to ℓ^2-RSC, SRSC captures the locally linear structure of the data manifold by encouraging nearby data to share dictionary atoms. In addition, SRSC preserves freedom in the spare representation of data without constraints on the magnitude of the sparse code, and it enjoys robustness to noise in the data.

The remainder of the chapter is organized as follows. SRSC and its optimiza-

tion algorithm, together with ℓ^2-RSC, are introduced in the next section. The theoretical property of the optimization of SRSC is shown in Section 5.3 with the theoretical guarantee on the obtained sub-optimal solution for each step of the coordinate descent for obtaining the support regularized sparse code. We then show the performance of the SRSC on data clustering, and conclude the chapter. We use bold letters for matrices and vectors, and regular lowercase letters for scalars throughout this chapter. A bold letter with superscript indicates the corresponding column of a matrix, and a bold letter with subscript indicates the corresponding element of a matrix or vector. $\|\cdot\|_F$ and $\|\cdot\|_p$ denote the Frobenius norm and the ℓ^p-norm, and $\mathrm{diag}(\cdot)$ indicates the diagonal elements of a matrix.

5.2 Support Regularized Sparse Coding

5.2.1 Capturing Local Linear Structure: Support Regularized Sparse Coding

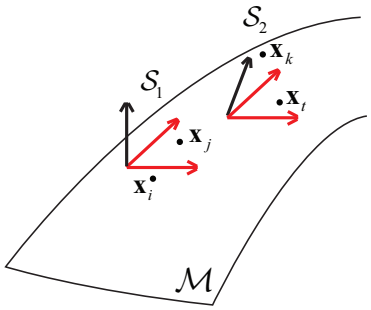


Figure 5.2: Illustration of capturing the locally linear structure of the data manifold by Support Regularized Sparse Coding. Nearby data are encouraged to share dictionary atoms. In this example, $\mathbf{x}_i$ and $\mathbf{x}_j$ choose three common dictionary atoms so they lie on or close to the local subspace $\mathcal{S}_1$ spanned by the common atoms, and it is the similar case for $\mathbf{x}_t$ and $\mathbf{x}_k$ with local subspace $\mathbf{S}_2$. Due to the smoothness of the support of the sparse codes, neighboring local subspaces, such as $\mathcal{S}_1$ and $\mathcal{S}_2$, can share dictionary atoms. In this example, the two local subspaces share two dictionary atoms marked in red.

In this section, we introduce Support Regularized Sparse Coding (SRSC) which is designed to capture the locally linear structure of the data manifold for sparse coding. One of the most important properties of manifold is that it is locally

Euclidean, and each data point in the manifold has a neighborhood that is homeomorphic to a Euclidean space. The success of several manifold learning methods, including LLE [50], SMCE [49] and Locally Linear Hashing [75], is built on exploiting the locally linear structure of manifold. The locally linear structure associated with each data point is a linear representation of that point by a set of its nearest neighbors in a nonparametric manner, from which the low-dimensional embedding complying to the manifold structure of the original data is obtained and used for various learning tasks. In the context of sparse coding, the data lie on or close to the subspace spanned by the dictionary atoms specified by nonzero elements of the corresponding sparse codes. Inspired by this observation, we propose to capture the locally linear structure of the data manifold for sparse coding by encouraging nearby data to share the atoms of the dictionary, so that nearby data are on or close to the local subspace spanned by the common dictionary atoms (see Figure 5.2).

In order to obtain the sparse codes with similar support and nonzero elements so as to capture the locally linear structure of the data manifold, we propose Support Regularized Sparse Coding (SRSC), which uses support distance to measure the distance between the sparse codes of nearby data. Given a proper symmetric similarity matrix $\mathbf{A}$, the sparse code $\mathbf{Z}$ that captures the locally linear structure of the manifold minimizes the following support regularization term:

$$\mathbf{R}_{\mathbf{A}}(\mathbf{Z}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{A}_{ij}d(\mathbf{Z}^i, \mathbf{Z}^j) \tag{5.4}$$

$\mathbf{A}$ is usually the adjacency matrix of K-nearest-neighbor (KNN) graph; i.e., $\mathbf{A}_{ij} = 1$ if and only if $\mathbf{x}_i$ is among the K nearest neighbors of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among the K nearest neighbors of $\mathbf{x}_i$. Note that KNN is extensively used in the manifold learning literature, such as Locally Linear Embedding (LLE) [50], Laplacian Eigenmaps [51] and Sparse Manifold Clustering and Embedding (SMCE) [49], to establish the local neighborhood in the manifold. d indicates the support distance. For two vectors $\mathbf{u}, \mathbf{v}$ of the same size, their support distance is defined below:

$$d(\mathbf{u}, \mathbf{v}) = \sum_{t=1}^{|\mathbf{u}|}(\mathbb{I}_{\mathbf{u}_t=0,\mathbf{v}_t\neq 0} + \mathbb{I}_{\mathbf{u}_t\neq 0,\mathbf{v}_t=0}) \tag{5.5}$$

When the support distance between $\mathbf{Z}^i$ and $\mathbf{Z}^j$ is small for nonzero $\mathbf{A}_{ij}$, $\mathbf{x}_i$ and $\mathbf{x}_j$ choose similar atoms of the dictionary for sparse representation. There-

fore, SRSC captures the locally linear structure of the data manifold by encouraging nearby data to share dictionary atoms, wherein the common atoms shared by nearby data serve as the basis of the locally linear space.

The optimization problem of SRSC is

$$\begin{aligned} &\min_{\mathbf{D},\mathbf{Z}} L(\mathbf{D},\mathbf{Z}) = \sum_{i=1}^{n} \frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\mathbf{Z}^i\|_2^2 + \lambda\|\mathbf{Z}^i\|_1 + \gamma\mathbf{R}_{\mathbf{A}}(\mathbf{Z}) \\ &s.t. \|\mathbf{D}^k\|_2 \leq 1, k = 1, \ldots, p \end{aligned} \tag{5.6}$$

where $\gamma > 0$ is the weighting parameter for the support regularization term. Similar to [76], the problem (5.6) is optimized alternatingly with respect to the dictionary $\mathbf{D}$ and the sparse code $\mathbf{Z}$ respectively with the other variable fixed.

Optimizing with respect to $\mathbf{D}$ with fixed $\mathbf{Z}$

The optimization with respect to $\mathbf{D}$ with fixed $\mathbf{Z}$ is a quadratic programming problem:

$$\begin{aligned} &\min_{\mathbf{D}} \frac{1}{2}\|\boldsymbol{X} - \mathbf{D}\mathbf{Z}\|_F^2 \\ &s.t. \|\mathbf{D}^k\|_2 \leq 1, k = 1, \ldots, p \end{aligned} \tag{5.7}$$

which can be solved using Lagrangian dual [76].

Optimizing with respect to $\mathbf{Z}$ with fixed $\mathbf{D}$

We use coordinate descent to optimize (5.6) with respect to $\mathbf{Z}$ with fixed $\mathbf{D}$:

$$\min_{\mathbf{Z}} \sum_{i=1}^{n} \frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\mathbf{Z}^i\|_2^2 + \lambda\|\mathbf{Z}^i\|_1 + \gamma\mathbf{R}_{\mathbf{A}}(\mathbf{Z}) \tag{5.8}$$

In each step of coordinate descent, the optimization is performed over the i-th column of $\mathbf{Z}$, while fixing all the other sparse codes $\{\mathbf{Z}^j\}_{j\neq i}$. For each $1 \leq i \leq n$, the optimization problem for $\mathbf{Z}^i$ is

$$\min_{\mathbf{Z}^i} F(\mathbf{Z}^i) = \frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\mathbf{Z}^i\|_2^2 + \lambda\|\mathbf{Z}^i\|_1 + \gamma\mathbf{R}_{\mathbf{A}}(\mathbf{Z}^i) \tag{5.9}$$

where $\mathbf{R_A}(\mathbf{Z}^i) = \sum_{j=1}^{n} \mathbf{A}_{ij} d(\mathbf{Z}^i, \mathbf{Z}^j)$.

Inspired by recent advances in solving non-convex optimization problems by proximal linearized method [52], iterative proximal gradient descent method (PGD) is used to optimize the nonconvex problem (5.9). Although the proximal mapping is typically associated with a lower semicontinuous function [52] and it can be verified that $\mathbf{R_A}$ is not always lower semicontinuous, we can still derive a PGD-style iterative method to optimize (5.9).

Define $\mathbf{G}^{\mathbf{A}} \in \mathbb{R}^{p \times n}$ as $\mathbf{G}^{\mathbf{A}}_{ki} = \sum_{j=1} \mathbf{A}_{ij} \mathbb{1}_{\mathbf{Z}_{kj}=0} - \sum_{j=1} \mathbf{A}_{ij} \mathbb{1}_{\mathbf{Z}_{kj}\neq 0}$ where $\mathbb{1}$ is the indicator function; then $\mathbf{G}^{\mathbf{A}}_{ki}$ indicates the degree to which $\mathbf{Z}_{ki}$ is encouraged to be nonzero and it can be verified that

$$\mathbf{R_A}(\mathbf{Z}^i) = \sum_{k=1}^{p} \mathbf{G}^{\mathbf{A}}_{ki} \mathbb{1}_{\mathbf{Z}_{ki}\neq 0} \tag{5.10}$$

Since each indicator function $\mathbb{1}_{\mathbf{Z}_{ki}\neq 0}$ is lower semicontinuous, $\mathbf{R_A}$ is lower semicontinuous if $\mathbf{G}^{\mathbf{A}}_{ki} \geq 0$ for $k = 1, \ldots, p$. In the following text, we let $Q(\mathbf{Z}^i) = \frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\mathbf{Z}^i\|_2^2$. The superscript with bracket indicates the iteration number of PGD or the iteration number of the coordinate descent without confusion. The PGD-style iterative method for optimizing (5.9) is

$$\tilde{\mathbf{Z}^i}^{(t)} = \mathbf{Z}^{i(t-1)} - \frac{1}{\tau s}(\mathbf{D}^\top \mathbf{D}\mathbf{Z}^{i(t-1)} - \mathbf{D}^\top \mathbf{x}_i) \tag{5.11}$$

$$\mathbf{Z}_{ki}{}^{(t)} = \begin{cases} \underset{v \in \{\mathbf{u}_k, 0\}}{\arg\min} H_k(v) & : \quad \mathbf{u}_k \neq 0 \text{ or } \mathbf{u}_k = 0 \text{ and } \mathbf{G}^{\mathbf{A}}_{ki} \geq 0 \\ \varepsilon & : \quad \mathbf{u}_k = 0 \text{ and } \mathbf{G}^{\mathbf{A}}_{ki} < 0 \end{cases} \tag{5.12}$$

for $k = 1, \ldots, p$ and ε is any real number such that $\varepsilon \neq 0$ and $H_k(\varepsilon) \leq H_k(\mathbf{Z}^{(t-1)}_{ki})$. H_k and $\mathbf{u}$ are defined as

$$H_k(v) = \frac{\tau s}{2}(v - \tilde{\mathbf{Z}}^{(t)}_{ki})^2 + \lambda|v| + \gamma \mathbf{G}^{\mathbf{A}}_{ki} \mathbb{1}_{v\neq 0} \tag{5.13}$$

for $v \in \mathbb{R}$ and each $1 \leq k \leq p$, and

$$\mathbf{u} = \max\{|\tilde{\mathbf{Z}^i}^{(t)}| - \frac{\lambda}{\tau s}, 0\} \circ \operatorname{sign}(\tilde{\mathbf{Z}^i}^{(t)}) \tag{5.14}$$

where $\circ$ means element-wise multiplication.

Proposition 2 shows that the PGD-style iterative method decreases the value of the objective function in each iteration.

Proposition 2. *Letting the sequence $\{\mathbf{Z}^{i(t)}\}_t$ be generated by the PGD-style iterative method with (5.11) and (5.12) in each $t \geq 1$, then the sequence of the objective $\{F(\mathbf{Z}^{i(t)})\}_t$ decreases, and the following inequality holds for $t \geq 1$:*

$$F(\mathbf{Z}^{i(t)}) \leq F(\mathbf{Z}^{i(t-1)}) - \frac{(\tau-1)s}{2}\|\mathbf{Z}^{i(t)} - \mathbf{Z}^{i(t-1)}\|_2^2 \tag{5.15}$$

And it follows that the sequence $\{F(\mathbf{Z}^{i(t)})\}_t$ converges.

Remark 2. *(5.11) and (5.12) in each iteration of the proposed PGD-style iterative method in Proposition 2 resemble that of the ordinary PGD. (5.11) performs gradient descent on the differential part, and (5.12) can be viewed as an approximate solution to the proximal mapping*

$$\min_{\mathbf{v}\in\mathbb{R}^p} H(\mathbf{v}) = \frac{\tau s}{2}\|\mathbf{v} - \tilde{\mathbf{Z}}^{i(t)}\|_2^2 + \lambda\|\mathbf{v}\|_1 + \gamma\mathbf{R}_{\mathbf{A}}(\mathbf{v}).$$

Since $\mathbf{R}_{\mathbf{A}}(\mathbf{Z}^i)$ is not always lower semicontinuous, $\arg\min_{\mathbf{v}\in\mathbb{R}^p} H(\mathbf{v})$ is not guaranteed to exist. One can see a simple example where this happens when $\mathbf{u}_k = 0$ and $\mathbf{G}^{\mathbf{A}}_{ki} < 0$ for some $k = 1, \ldots, p$, and in this case $\inf_{v\in\mathbb{R}} H_k(v) = \mathbf{G}^{\mathbf{A}}_{ki}$ but this infinitum cannot be achieved.

In (5.11), $\tau > 1$ is a constant and s is the Lipschitz constant for the gradient of function $Q(\cdot)$, namely

$$\|\nabla Q(\mathbf{y}) - \nabla Q(\mathbf{z})\|_2 \leq s\|\mathbf{y} - \mathbf{z}\|_2, \ \forall\, \mathbf{y}, \mathbf{z} \in \mathbb{R}^p \tag{5.16}$$

The PGD-style iterative method starts from $t = 1$ and continues until the sequence $\{F(\mathbf{Z}^{i(t)})\}$ converges or maximum iteration number is achieved. When the proximal method converges or terminates for each $\mathbf{Z}^i$, the step of coordinate descent for $\mathbf{Z}^i$ is finished and the optimization algorithm proceed to optimize other sparse codes.

Algorithm 3 describes the algorithm of SRSC. We solve the ordinary sparse coding problem (5.1) by the online dictionary learning method ([77]) and use the dictionary and the sparse code as the initialization $\mathbf{D}^{(0)}$ and $\mathbf{Z}^{(0)}$ for the alternating method in Algorithm 3.

Algorithm 3 Support Regularized Sparse Coding

Input:

The data set $\boldsymbol{X} = \{\mathbf{x}_i\}_{i=1}^n$, the parameter λ, γ, maximum iteration number M for the alternating method over $\mathbf{D}$ and $\mathbf{Z}$, and maximum iteration number M_z for coordinate descent on $\mathbf{Z}$, maximum iteration number M_p for the PGD-style iterative method on each $\mathbf{Z}^i$ (i = 1,. . . ,n).
and stopping threshold ε.

1: $m = 0$
2: **while** $m \leq M$ **do**
3: Perform coordinate descent to optimize (5.8) and obtain $\mathbf{Z}^{(m)}$ with fixed $\mathbf{D}^{(m-1)}$. In i-th ($1 \leq i \leq n$) step of each iteration of coordinate descent, solve (5.9) using the PGD-style iterative method (5.11) and (5.12) to update $\mathbf{Z}^i$ in each iteration of the PGD-style iterative method.
4: **if** $|L(\mathbf{D}^{(m)}, \mathbf{Z}^{(m)}) - L(\mathbf{D}^{(m-1)}, \mathbf{Z}^{(m-1)})| < \varepsilon$ **then**
5: **break**
6: **else**
7: $m = m + 1$.
8: **end if**
9: **end while**

Output: the support regularized sparse code $\hat{\mathbf{Z}}$ when the above iterations converge or maximum iteration number is achieved.

5.2.2 Related work: ℓ^2 Regularized Sparse Coding

The manifold assumption [5] is usually employed by existing regularized sparse coding methods [6, 7, 8, 9] to obtain the sparse code according to the manifold structure of the data. Interpreting the sparse code of a data point as its embedding, the manifold assumption in the case of sparse coding for most existing methods requires that if two points $\mathbf{x}_i$ and $\mathbf{x}_j$ are close in the intrinsic geometry of the submanifold, their corresponding sparse codes $\mathbf{Z}^i$ and $\mathbf{Z}^j$ are also expected to be similar to each other in the sense of ℓ^2-distance [8, 9]. In other words, $\mathbf{z}$ varies smoothly along the geodesics in the intrinsic geometry. Based on the spectral graph theory [39], extensive literature uses graph Laplacian to impose local smoothness of the embedding and preserve the local manifold structure [5, 8, 9].

The sparse code $\mathbf{Z}$ that captures the local geometric structure of the data in accordance with the manifold assumption by graph Laplacian minimizes the following ℓ^2 regularization term:

$$\mathbf{R}_{\mathbf{A}}^{(\ell^2)}(\mathbf{Z}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{A}_{ij}\|\mathbf{Z}^i - \mathbf{Z}^j\|_2^2 \tag{5.17}$$

where the ℓ^2-norm is used to measure the distance between sparse codes, and $\mathbf{A}$ is the same as that in Section 5.2.1. $\mathbf{L_A} = \mathbf{D_A} - \mathbf{A}$ is the graph Laplacian associated with the similarity matrix $\mathbf{A}$, the degree matrix $\mathbf{D_A}$ is a diagonal matrix with each diagonal element being the sum of the elements in the corresponding row of $\mathbf{S}$, namely $(\mathbf{D_A})_{ii} = \sum_{j=1}^{n} \mathbf{A}_{ij}$. To the best of our knowledge, such ℓ^2 regularization is employed by most methods that use graph regularization for sparse coding. Incorporating the ℓ^2 regularization term into the optimization problem of sparse coding (5.1), the formulation of ℓ^2 Regularized Sparse Coding (ℓ^2-RSC) is

$$\min_{\mathbf{Z}} L^{(\ell^2)}(\mathbf{Z}) = \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{D}\mathbf{Z}^i\|_2^2 + \lambda\|\mathbf{Z}^i\|_1 + \gamma^{(\ell^2)}\mathbf{R}_{\mathbf{A}}^{(\ell^2)}(\mathbf{Z}) \tag{5.18}$$

Although ℓ^2-RSC imposes the local smoothness on the sparse codes, it does not capture the locally linear structure of the data manifold. By promoting the smoothness on the support of the sparse codes rather than their ℓ^2-distance, SRSC encodes the locally linear structure of the manifold in the sparse codes while reserving freedom in the sparse representation of the data with no constraints on the magnitude of the sparse codes. Moreover, as pointed out by [78], support regularization offers robustness to noise for sparse coding. In SRSC, all the data consult their neighbors for choosing the dictionary atoms rather than choosing the atoms on their own, and the sparse codes of the noisy data are suppressed since they are forced to choose similar or the same atoms as the nearby clean data instead of choosing the atoms in the interests of representing themselves.

5.3 Theoretical Analysis

It can be observed that optimization by coordinate descent over the sparse code in Section 5.2.1 is important for the overall optimization of SRSC, and each step of the coordinate descent (5.9) is a difficult nonconvex problem and crucial for obtaining the support regularized sparse code, where the nonconvexity comes from the support regularization term $\mathbf{R_A}(\mathbf{Z}^i)$ (5.10). Therefore, the optimization of (5.9) plays an important role in the overall optimization of SRSC. In the previous section, a PGD-style iterative method is proposed to decrease the value of the objective in each iteration. In this section, we provide further theoretical analysis on the optimization of problem (5.9) when $\mathbf{G}_{ki}^{\mathbf{A}} \geq 0$ for $k = 1, \ldots, p$. This condition

is equivalent to the condition that the support regularization function

$$\mathbf{R}_{\mathbf{c}}(\mathbf{v}) \triangleq= \sum_{k=1}^{p} \mathbf{c}_k \mathbb{I}_{\mathbf{v}_k \neq 0} \tag{5.19}$$

is lower semicontinuous, where $\mathbf{c} \in \mathbb{R}^p$ is the coefficients and $\mathbf{c}_k = \mathbf{G}^{\mathbf{A}}_{ki}$. We prove that the sequence $\{\mathbf{Z}^{i(t)}\}_t$ produced by PGD converges to the sub-optimal solution which is a critical point of the objective (5.9). By connecting the support regularized function to the capped-ℓ^1 norm and the nonconvexity analysis of the support regularization term, we present the bound for ℓ^2-distance between the sub-optimal solution and the global optimal solution to (5.9) in Theorem 10. Note that our analysis is valid for all $1 \leq i \leq n$.

We first have the following result that the support regularization function (5.19) is lower semicontinuous if and only if all the coefficients $\mathbf{c}$ are nonnegative.

Proposition 3. *The support regularization function (5.19) is lower semicontinuous if and only if all the coefficients* $\mathbf{c}$ *are nonnegative.*

Therefore, if $\mathbf{G}^{\mathbf{A}}_{ki} \geq 0$ for $k = 1, \ldots, p$, the support regularization term $\mathbf{R}_{\mathbf{A}}(\mathbf{Z}^i)$ is lower semicontinuous with respect to $\mathbf{Z}^i$ in (5.10). In this case, the PGD-style iterative method proposed in Section 5.2.1 for each iteration $t \geq 1$ becomes

$$\tilde{\mathbf{Z}^i}^{(t)} = \mathbf{Z}^{i(t-1)} - \frac{1}{\tau s}(\mathbf{D}^\top \mathbf{D}\mathbf{Z}^{i(t-1)} - \mathbf{D}^\top \mathbf{x}_i) \tag{5.20}$$

$$\mathbf{Z}_{ki}{}^{(t)} = \underset{v \in \{\mathbf{u}_k, 0\}}{\arg\min} H_k(v), k = 1, \ldots, p \tag{5.21}$$

which is equivalent to the updates rules in ordinary proximal gradient descent method. In the following lemma, we show that the sequence $\{\mathbf{Z}^{i(t)}\}_t$ generated by (5.20) and (5.21) converges to a critical point of $F(\mathbf{Z}^i)$, denoted by $\hat{\mathbf{Z}}^i$. Denote by $\mathbf{Z}^{i*}$ the global optimal solution to the original optimization problem (5.9). The following lemma also shows that both $\hat{\boldsymbol{\alpha}}^i$ and $\boldsymbol{\alpha}^{i*}$ are local solutions to the capped-ℓ^1 regularized problem (5.22). Before stating the lemma, the following definitions are introduced which are essential for our analysis.

Definition 6. *(Critical points) Given the non-convex function* $f \colon \mathbb{R}^n \to R \cup \{+\infty\}$ *which is a proper and lower semi-continuous function.*

- *For a given* $\mathbf{x} \in \mathrm{dom} f$*, its Frechet subdifferential of* f *at* $\mathbf{x}$*, denoted by* $\tilde{\partial} f(x)$*, is the set of all vectors* $\mathbf{u} \in \mathbb{R}^n$ *which satisfy*

$$\limsup_{\mathbf{y} \neq \mathbf{x}, \mathbf{y} \to \mathbf{x}} \frac{f(\mathbf{y}) - f(\mathbf{x}) - \langle \mathbf{u}, \mathbf{y} - \mathbf{x} \rangle}{\|\mathbf{y} - \mathbf{x}\|} \geq 0$$

- *The limiting-subdifferential of* f *at* $\mathbf{x} \in \mathbb{R}^n$*, denoted by written* $\partial f(x)$*, is defined by*

$$\begin{aligned} \partial f(x) = \{ & \mathbf{u} \in \mathbb{R}^n \colon \exists \mathbf{x}^k \to \mathbf{x}, f(\mathbf{x}^k) \to f(\mathbf{x}), \\ & \tilde{\mathbf{u}}^k \in \tilde{\partial} f(\mathbf{x}_k) \to \mathbf{u} \} \end{aligned}$$

The point $\mathbf{x}$ *is a critical point of* f *if* $0 \in \partial f(x)$.

Also, we are considering the following capped-ℓ^1 regularized problem, which replaces the indicator function in the support regularization term $\mathbf{R}_{\mathbf{A}}(\mathbf{Z}^i)$ with the continuous capped-ℓ^1 regularization term $\mathbf{T}$:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^n} L_{\mathrm{capped}-\ell^1}(\boldsymbol{\beta}) = \frac{1}{2} \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 + \mathbf{T}(\boldsymbol{\beta}; b) \tag{5.22}$$

where $\mathbf{T}(\boldsymbol{\beta}; b) = \sum_{k=1}^{p} T_k(\boldsymbol{\beta}_k; b)$, $T_k(t; b) = \gamma \mathbf{G}_{ki}^{\mathbf{A}} \frac{\min\{|t|, b\}}{b}$ for some $b > 0$. It can be seen that the objective function of the capped-ℓ^1 problem approaches that of (5.9) when $\frac{\min\{|t|, b\}}{b}$ approaches the indicator function $\mathbb{1}_{t \neq 0}$, as $b \to 0+$. Defining $\mathbf{P}(\cdot; b) = \lambda \|\cdot\|_1 + \mathbf{T}(\cdot; b)$, the location solution to the capped-ℓ^1 problem is defined as follows:

Definition 7. *(Local solution) A vector* $\tilde{\boldsymbol{\beta}}$ *is a local solution to the problem (5.22) if*

$$\|\mathbf{D}^\top (\mathbf{D}\tilde{\boldsymbol{\beta}} - \mathbf{x}_i) + \dot{\mathbf{P}}(\tilde{\boldsymbol{\beta}}; b)\|_2 = 0 \tag{5.23}$$

where $\dot{\mathbf{P}}(\tilde{\boldsymbol{\beta}}; b) = [\dot{P}_1(\tilde{\boldsymbol{\beta}}_1; b), \dot{P}_2(\tilde{\boldsymbol{\beta}}_2; b), \ldots, \dot{P}_p(\tilde{\boldsymbol{\beta}}_p; b)]^\top$*,* $P_k(t; b) = \lambda |t| + T_k(t; b)$ *for* $k = 1, \ldots, p$.

Note that in the above definition and the following text, $\dot{P}_k(t; b)$ can be chosen as any value between the right differential $\frac{\partial P_k}{\partial t}(t+; b)$ (or $\dot{P}_k(t+; b)$) and left differential $\frac{\partial P_k}{\partial t}(t-; b)$ (or $\dot{P}_k(t-; b)$) for $k = 1, \ldots, p$.

Definition 8. *(Degree of Nonconvexity of a Regularizer) For $\kappa \geq 0$ and $t \in \mathbb{R}$, define*

$$\theta(t, \kappa) := \sup_s \{-\mathrm{sgn}(s-t)(\dot{P}(s; b) - \dot{P}(t; b)) - \kappa|s - t|\}$$

as the degree of nonconvexity for function P. If $\mathbf{u} = (u_1, \ldots, u_p)^\top \in \mathbb{R}^p$, $\theta(\mathbf{u}, \kappa) = [\theta(u_1, \kappa), \ldots, \theta(u_p, \kappa)]$. sgn is a sign function defined as

$$\mathrm{sgn}(x) = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases}$$

Note that $\theta(t, \kappa) = 0$ for convex function P.

Let $\hat{\mathbf{S}}_i = \mathrm{supp}(\hat{\mathbf{Z}}^i)$, $\mathbf{Z}^{i*}$ be the globally optimal solution to (5.9), and $\mathbf{S}_i^* = \mathrm{supp}(\mathbf{Z}^{i*})$.

Lemma 6. *For any $1 \leq i \leq n$, if $\mathbf{G}_{ki}^{\mathbf{A}} \geq 0$ for $k = 1, \ldots, p$, the sequence $\{\mathbf{Z}^{i(t)}\}_t$ generated by (5.11) and (5.12) converges to a critical point of $F(\mathbf{Z}^i)$, which is denoted by $\hat{\mathbf{Z}}^i$. Moreover, if*

$$0 < b < \min\{\min_{j \in \hat{\mathbf{S}}_i} |\hat{\boldsymbol{\alpha}}_j^i|, \max_{k \notin \hat{\mathbf{S}}_i, \mathbf{G}_{ki}^{\mathbf{A}} \neq 0} \frac{\gamma \mathbf{G}_{ki}^{\mathbf{A}}}{(\frac{\partial Q}{\partial \boldsymbol{\alpha}_k^i}|_{\boldsymbol{\alpha}^i = \hat{\boldsymbol{\alpha}}^i} - \lambda, 0)_+}, \tag{5.24}$$

$$\min_{j \in \mathbf{S}_i^*} |\boldsymbol{\alpha}_j^{i*}|, \max_{k \notin \mathbf{S}_i^*, \mathbf{G}_{ki}^{\mathbf{A}} \neq 0} \frac{\gamma \mathbf{G}_{ki}^{\mathbf{A}}}{(\frac{\partial Q}{\partial \boldsymbol{\alpha}_j^i}|_{\boldsymbol{\alpha}^i = \boldsymbol{\alpha}^{i*}} - \lambda, 0)_+}\}$$

(if the denominator is 0, $\frac{\cdot}{0}$ is defined to be $+\infty$ in the above inequality), then both $\hat{\boldsymbol{\alpha}}^i$ and $\boldsymbol{\alpha}^{i*}$ are local solutions to the capped-ℓ^1 regularized problem (5.22).

Using the degree of nonconvexity of the regularizer $\mathbf{P}$, we have the following theorem showing that the sub-optimal solution $\hat{\mathbf{Z}}^i$ obtained by our proximal method is close to the globally optimal solution to the original problem (5.9), i.e. $\mathbf{Z}^{i*}$.

Theorem 7. *(Sub-optimal solution is close to the globally optimal solution) For any $1 \leq i \leq n$, let $\mathbf{E}_i = \hat{\mathbf{S}}_i \cup \mathbf{S}_i^*$, and suppose $\mathbf{D}_{\mathbf{E}_i}$ is not singular with $\kappa_0 \triangleq \sigma_{\min}(\mathbf{D}_{\mathbf{E}_i}) > 0$. When $\kappa_0^2 > \kappa > 0$ and b is chosen according to (5.24) as in Lemma 8, let $\tilde{\mathbf{S}}_i = (\hat{\mathbf{S}}_i \setminus \mathbf{S}_i^*) \cup (\mathbf{S}_i^* \setminus \hat{\mathbf{S}}_i)$ be the symmetric difference between $\hat{\mathbf{S}}_i$*

and $\mathbf{S}_i^*$*; then*

$$\|\Delta\|_2 \leq \frac{1}{\kappa_0^2 - \kappa}\Big(\sum_{k \in \tilde{\mathbf{S}}_i \cap \hat{\mathbf{S}}_i} (\max\{0, \frac{\gamma \mathbf{G}^{\mathbf{A}}_{ki}}{b} - \kappa|t-b|\})^2 + \sum_{k \in \tilde{\mathbf{S}}_i \setminus \hat{\mathbf{S}}_i} (\max\{0, \frac{\gamma \mathbf{G}^{\mathbf{A}}_{ki}}{b} - \kappa b\})^2 \Big)^{\frac{1}{2}} \tag{5.25}$$

Remark 3. *Note that the bound for distance between the sub-optimal solution and the globally optimal solution presented in Theorem 10 does not require typical RIP conditions. Also, when* $\frac{\gamma \mathbf{G}^{\mathbf{A}}_{ki}}{b} - \kappa|t-b|$ *and* $\frac{\gamma \mathbf{G}^{\mathbf{A}}_{ki}}{b} - \kappa b$ *are no greater than* 0*, or they are small positive numbers, the sub-optimal solution* $\hat{\boldsymbol{\alpha}}^i$ *is equal to or very close to the globally optimal solution.*

5.4 Deep Support Regularized Sparse Coding

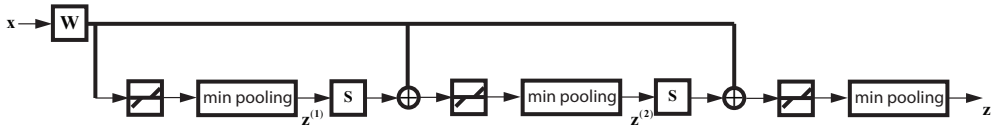


Figure 5.3: Illustration of Deep-SRSC for approximate Support Regularized Sparse Coding.

Inspired by LISTA network and the PGD-style iterative method (5.11) and (5.12) for SRSC, we propose Deep Support Regularized Sparse Coding (Deep-SRSC) illustrated in Figure 5.3, which is a neural network that produces the approximate support regularized sparse code for SRSC. Letting $\mathbf{W} = \frac{1}{L}\mathbf{D}^\top$, $\mathbf{S} = \mathbf{I} - \frac{1}{L}\mathbf{D}^\top\mathbf{D}$ where $L = \tau s$, then each stage in the recurrent structure of Deep-SRSC implements one iteration of PGD-style iterative method, i.e. (5.11) and (5.12). $\mathbf{W}$, $\mathbf{S}$ and L are to be learned by the network rather than computed from a pre-computed dictionary $\mathbf{D}$, and $\mathbf{S}$ is shared over different layers. The min-pooling neuron in Deep-SRSC outputs the result of $\arg\min_{v \in \{\mathbf{u}_k, 0\}} H_k(v)$ or ε, according to the update rule (5.12). Figure 5.3 shows Deep-SRSC with 3 layers.

Denote the training data by $\mathbf{x}_1, \ldots, \mathbf{x}_m$, and let $\mathbf{Z}_{\text{sr}}$ be the ground truth support regularized sparse codes of the training data which are obtained by the optimization method introduced in the previous section. Let f_{sr} be the Deep-SRSC encoder which produces the approximate support regularized sparse code $\mathbf{z} = f_{\text{sr}}(\mathbf{x}, \Theta_{\text{sr}})$,

where $\Theta_{\rm sr} = (\mathbf{W}, \mathbf{S}, L)$ denotes the parameters of Deep-SRSC. Then the parameters of Deep-SRSC are learned by minimizing the following cost function which measures the distance between the predicted approximate support regularized sparse codes and the ground truth ones: $\frac{1}{m}\sum_{i=1}^{m} \|\mathbf{Z}_{\rm sr}^i - f_{sr}(\mathbf{x}_i, \Theta_{\rm sr})\|_2^2$. Similar to the LISTA network, the above optimization is performed by stochastic gradient descent and back-propagation.

In the experimental results shown in the next section, Deep-SRSC with different number of layers are employed to produce the approximate support regularized sparse code.

Table 5.1: Clustering Results on USPS Handwritten Digits Database

USPS / # Clusters	Measure	KM	SC	Sparse Coding	ℓ^2-RSC	SRSC
c = 4	AC	0.9243	0.4514	0.9869	0.9869	**0.9880**
	NMI	0.7782	0.4160	0.9429	0.9429	**0.9467**
c = 6	AC	0.7130	0.4325	0.7781	0.7781	**0.9723**
	NMI	0.6845	0.4865	0.8507	0.8507	**0.9135**
c = 8	AC	0.7294	0.4227	0.8163	0.8163	**0.9645**
	NMI	0.6851	0.4811	0.8669	0.8669	**0.9027**
c = 10	AC	0.6878	0.4041	0.8178	0.8287	**0.8293**
	NMI	0.6312	0.4765	0.8321	0.8398	**0.8471**

Table 5.2: Clustering Results on Various Data Sets

Data Set	Measure	KM	SC	Sparse Coding	ℓ^2-RSC	SRSC
COIL-20	AC	0.6274	0.3347	0.9903	0.9903	**0.9944**
	NMI	0.7533	0.5667	0.9879	0.9879	**0.9933**
COIL-100	AC	0.5221	0.2372	0.6979	0.6979	**0.7267**
	NMI	0.7633	0.5410	0.8837	0.8837	**0.8876**
UCI Gesture Phase Segmentation	AC	0.3868	0.3375	0.4003	0.4023	**0.4123**
	NMI	0.1191	**0.1300**	0.1164	0.1164	0.1187

Table 5.3: Prediction Error (Squared Error Between the Predicted Codes and the Ground Truth Codes) of Deep-SRSC with Different Layers

Deep-SRSC	1-layer	2-layer	6-layer
Error	0.14	0.09	0.07

Table 5.4: Clustering Results on the Test Data of USPS Data Set

Measure	KM	SC	Sparse Coding	ℓ^2-RSC	SRSC	6-layer Deep-SRSC
AC	0.6020	0.3279	0.6408	0.6462	**0.7225**	0.7000
NMI	0.5522	0.4372	0.7011	0.7011	0.7045	0.6817

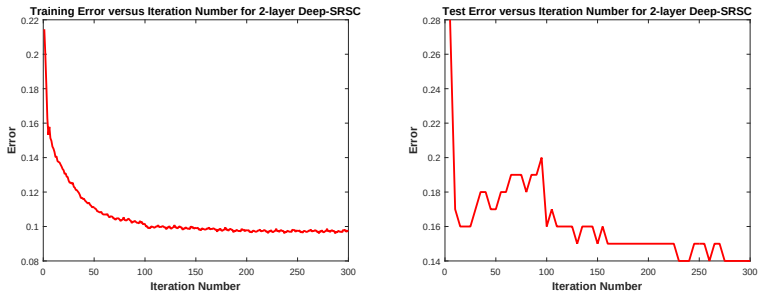


Figure 5.4: Training and test error for Deep-SRSC with 1 layers.

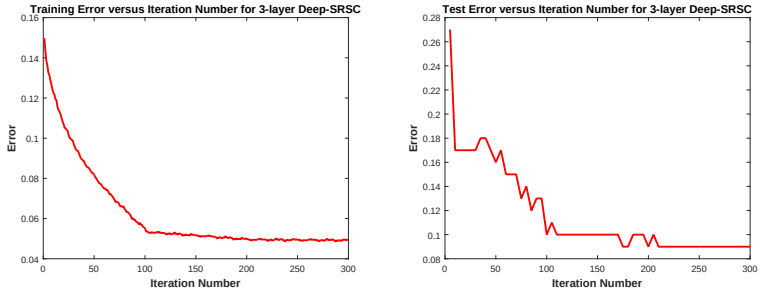


Figure 5.5: Training and test error for Deep-SRSC with 2 layers.

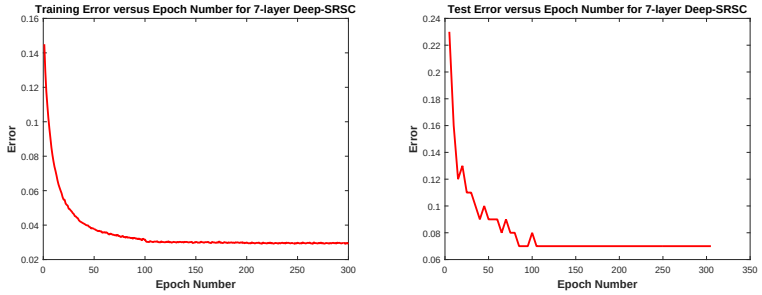


Figure 5.6: Training and test error for Deep-SRSC with 6 layers.

5.5 Experimental Results

5.5.1 Clustering Performance

In this subsection, the superiority of SRSC is demonstrated by its performance in data clustering on various data sets, e.g. USPS handwritten digits data set, COIL-20, COIL-100 and UCI Gesture Phase Segmentation data set. Two measures are used to evaluate the performance of the clustering methods, i.e. the Accuracy

(AC) and the Normalized Mutual Information (NMI) [53]. SRSC is compared to K-means (KM), Spectral Clustering (SC), and ℓ^2-RSC in Section 5.2.2. We set $K = 3$ for building the adjacency matrix $\mathbf{A}$ of KNN graph for both ℓ^2-RSC and SRSC, dictionary size $p = 300$, and set $\gamma^{(\ell^2)} = 1$ which is the suggested default value in [8]. The default value for the support regularization term for SRSC is $\gamma = 0.5$. SRSC is implemented by both MATLAB and CUDA C++ with extreme efficiency.

The USPS handwritten digits data set is comprised of $n = 9298$ handwritten images of nine digits $0 - 9$, and each image is of size 16×16 and represented by a 256-dimensional vector. The whole data set is divided into training set of 7291 images and test set of 2007 images. We run Algorithm 3 to obtain the support regularized sparse code $\hat{\mathbf{Z}}$, then build a $n \times n$ similarity matrix $\mathbf{Y}$ over all the data. Two similarity measure are employed: the first similarity is the positive part of the inner product of their corresponding sparse codes, namely $\mathbf{Y}_{ij} = \max\{0, \hat{\mathbf{Z}}^{i\top}\hat{\mathbf{Z}}^{j}\}$, the second one is $\mathbf{Y}_{ij} = \mathbf{A}_{ij}\mathbf{q}_{\hat{\mathbf{Z}}^i}^{\top}\mathbf{q}_{\hat{\mathbf{Z}}^j}$ where $\mathbf{q}_{\mathbf{v}}$ is a binary vector of the same size as $\mathbf{v}$ with element 1 at the indices of nonzero elements of $\mathbf{v}$. The second similarity measure considers the number of common dictionary atoms chosen by the sparse codes. Spectral clustering is performed on the similarity matrix $\mathbf{Y}$ to obtain the clustering result of SRSC, and the best performance among the two similarity measures is reported. The same procedure is performed by all the other sparse coding based methods to obtain clustering results. The clustering results of various methods are shown in Table 5.1. c in the left column of Table 5.1 is the cluster number, i.e. the first c clusters of the entire data are used for clustering.

COIL-20 Database has 1440 images of resolution 32×32 for 20 objects, and the background is removed in all images. The dimension of this data is 1024. Its enlarged version, COIL-100 Database, contains 100 objects with 72 images of resolution 32×32 for each object. The images of each object were taken 5 degrees apart when each object was rotated on a turntable. The UCI Gesture Phase Segmentation data set contains the gesture information of three users when they told stories of some comic strips in front of the Microsoft Kinect sensor. We use the processed file provided by the original data consisting of 9873 frames, and the gesture information in each frame is the vectorial velocity and acceleration of left hand, right hand, left wrist, and right wrist, represented by a 32-dimensional vector. The clustering results on these three data sets are shown in Table 5.2.

It can be observed from Tables 5.1 and 5.2 that SRSC always produces better clustering accuracy than other competing methods, due to its capability of captur-

ing the locally linear manifold structure of the data.

5.5.2 Approximation by Deep-SRSC

In this subsection, Deep-SRSC is employed as a fast encoder to approximate the support regularized sparse code of SRSC on the USPS data set. We adopt three settings wherein Deep-SRSC has 1 layers, 2 layers, and 6 layers respectively. The training and test error of Deep-SRSC with respect to the epoch number for different number of layers are shown in Figures 5.4, 5.5 and 5.6. We first run SRSC on the training set of USPS data to obtain the dictionary $\mathbf{D}_{\mathrm{sr}}$ and the support regularized sparse code $\mathbf{Z}_{\mathrm{sr}}$. Then the optimization problem (5.8) is solved by the PGD-style iterative method in Section 5.2.1, where $\boldsymbol{X}$ is the test data and $\mathbf{A}$ is the adjacency matrix of the KNN graph over the test data, to obtain the support regularized sparse code $\mathbf{Z}_{\mathrm{sr,test}}$ of the test data with dictionary $\mathbf{D}_{\mathrm{sr}}$. $\mathbf{Z}_{\mathrm{sr}}$ is used as the ground truth support regularized sparse code to train Deep-SRSC.

Table 5.3 shows the squared error between the predicted support regularized sparse codes and the ground truth codes of the test data, i.e. $\mathbf{Z}_{\mathrm{sr,test}}$. It can be observed that Deep-SRSC with more layers demonstrates smaller prediction error due to its better representation capability. Moreover, the codes predicted by 6-layer Deep-SRSC are used to perform clustering on the test data, with comparison to the performance of sparse coding and ℓ^2-RSC shown in Table 5.4. For either sparse coding or ℓ^2-RSC, the dictionary is first learned on the training data, then the sparse codes of the test data are obtained with respect to that dictionary. We can see that SRSC and its approximation, 6-layer Deep-SRSC, achieves the highest accuracy and NMI respectively.

5.6 Conclusion

We propose Support Regularized Sparse Coding (SRSC) which exploits the locally linear manifold structure for high-dimensional data while performing sparse coding, and SRSC achieves this goal by encouraging nearby data in the manifold to share dictionary atoms through a support regularization term in the formulation of regular sparse coding. Similar to LISTA which is a fast encoder for sparse coding, we also propose Deep-SRSC, a feed-forward neural network, as a fast encoder to approximate the support regularized sparse code produce by SRSC.

Experimental results demonstrate the effectiveness of SRSC by its application to data clustering. We also show that Deep-SRSC renders the approximate codes for SRSC with low prediction error, and the approximate codes generated by 6-layer Deep-SRSC also deliver compelling empirical performance on data clustering.

APPENDIX SUPPLEMENTARY DOCUMENTS FOR CHAPTER 2 AND CHAPTER 4

A.1 Supplementary Document for On a Theory of Nonparametric Pairwise Similarity for Clustering: Connecting Clustering to Classification

A.1.1 Proofs of Theorems and Lemmas

We provide detailed proofs of the theorems and lemmas. As stated in the Chapter 2, we define

$$f_0 \triangleq \sum_{i=1}^{Q} \pi^{(i)} f_{\max}^{(i)} \quad \sigma_0^2 \triangleq \|K\|_2^2 f_0 \tag{A.1}$$

Let $L, C > 0$ be constants which only depend on the VC characteristics of the Gaussian kernel K. For all $\lambda \geq C$ and $\sigma > 0$, we define

$$E_{\sigma^2} \triangleq \frac{\log\left(1 + \lambda/4L\right)}{\lambda L \sigma^2} \tag{A.2}$$

Lemma. *For any $P_{XY} \in \mathcal{P}_{XY}$, there exists a n_0 which depends on σ_0 and VC characteristics of K such that when $n > n_0$, with probability greater than $1 - 2QLh_n^{E_{\sigma_0^2}}$, the generalization error of the plug-in classifier satisfies*

$$R\left(\mathrm{PI}_S\right) \leq R_n^{\mathrm{PI}} + \mathcal{O}\Big(\sqrt{\frac{\log h_n^{-1}}{n h_n^d}} + h_n^\gamma\Big) \tag{A.3}$$

$$R_n^{\mathrm{PI}} = \sum_{i,j=1,\ldots,Q, i \neq j} \mathbb{E}_X\left[\hat{\eta}_{n,h_n}^{(i)}(X)\, \hat{\eta}_{n,h_n}^{(j)}(X)\right] \tag{A.4}$$

where E_{σ^2} is defined by (A.2), h_n is chosen such that $h_n \to 0, \frac{\log h_n^{-1}}{nh_n^d} \to 0$, $\hat{\eta}_{n,h_n}^{(i)}$ is the kernel estimator of the regression function. Moreover, the equality in (A.3) holds when $\hat{\eta}_{n,h_n}^{(i)} \equiv \frac{1}{Q}$ for $1 \leq i \leq Q$.

Theorem. *(Error of the Plug-In Classifier) Given the classification model $M_{\mathcal{Y}} = \left(\mathcal{S}, P_{XY}, \{\pi_i, f_i\}_{i=1}^{Q}, \mathrm{PI}\right)$ with $P_{XY} \in \mathcal{P}_{XY}$, there exists a n_1 which depends on σ_0, σ_1 and the VC characteristics of K such that when $n > n_1$, with probability greater than $1 - 2QLh_n^{E_{\sigma_0^2}} - L(\sqrt{2}h_n)^{E_{\sigma_0^2}} - QLh_n^{E_{\sigma_1^2}}$, the generalization error of the plug-in classifier satisfies*

$$R\left(\mathrm{PI}_S\right) \leq \hat{R}_n\left(\mathrm{PI}_S\right) + \mathcal{O}\Big(\sqrt{\frac{\log h_n^{-1}}{nh_n^d}} + h_n^{\gamma}\Big) \tag{A.5}$$

where $\hat{R}_n\left(\mathrm{PI}_S\right) = \frac{1}{n^2}\sum_{l,m}\theta_{lm}G_{lm,\sqrt{2}h_n}$, $\sigma_1^2 = \frac{\|K\|_2^2 f_{\max}}{f_{\min}}$, $\theta_{lm} = \mathbb{1}_{\{\mathbf{y}_l \neq \mathbf{y}_m\}}$ is a class indicator function and

$$G_{lm,h} = G_h\left(\mathbf{x}_l, \mathbf{x}_m\right), \; G_h\left(x, y\right) = \frac{K_h\left(x - y\right)}{\hat{f}_{n,h}^{\frac{1}{2}}\left(x\right)\hat{f}_{n,h}^{\frac{1}{2}}\left(y\right)} \tag{A.6}$$

E_{σ^2} is defined by (A.2), h_n is chosen such that $h_n \to 0, \frac{\log h_n^{-1}}{nh_n^d} \to 0$, $\hat{f}_{n,h_n}$ is the kernel density estimator of f defined by (2.3).

Corollary. *Under the assumption of Theorem 1, for any kernel bandwidth sequence $\{h_n\}_{n=1}^{\infty}$ such that $\lim_{n\to\infty} h_n = 0$ and $h_n > n^{-\alpha}$ where $0 < \alpha < \frac{1}{2d+2}$, with probability 1,*

$$\lim_{n\to\infty} \frac{\sqrt{\pi}}{2h_n}\hat{R}_n\left(\mathrm{PI}_S\right) = \int_S f\left(s\right)ds \tag{A.7}$$

Theorem. *(Error of the NN)*
Suppose the classification model is given as $M_{\mathcal{Y}} = \left(\mathcal{S}, P_{XY}, \{\pi_i, f_i\}_{i=1}^{Q}, \mathrm{NN}\right)$ with $P_{XY} \in \mathcal{P}_{XY}$ and the support of P_X is bounded by $[-M_0, M_0]^d$, there exists a n_0 which depends on σ_0 and VC characteristics of K such that when $n > n_0$, with probability greater than $1 - 2QLh_n^{E_{\sigma_0^2}} - (2M_0)^d n^{dd_0} e^{-n^{1-dd_0} f_{\min}}$, the generalization error of the NN satisfies:

$$R\left(\mathrm{NN}_S\right) \leq \hat{R}_n\left(\mathrm{NN}_S\right) + c_0\left(\sqrt{d}\right)^{\gamma} n^{-d_0\gamma} + \mathcal{O}\Big(\sqrt{\frac{\log h_n^{-1}}{nh_n^d}} + h_n^{\gamma}\Big) \tag{A.8}$$

where $\hat{R}_n(\mathrm{NN}) = \frac{1}{n}\sum_{1\le l<m\le n} H_{lm,h_n}\theta_{lm}$,

$$H_{lm,h_n} = K_{h_n}(\mathbf{x}_l - \mathbf{x}_m)\Big(\frac{\int_{\mathcal{V}_l}\hat{f}_{n,h_n}(x)\,\mathrm{d}x}{\hat{f}_{n,h_n}(\mathbf{x}_l)} + \frac{\int_{\mathcal{V}_m}\hat{f}_{n,h_n}(x)\,\mathrm{d}x}{\hat{f}_{n,h_n}(\mathbf{x}_m)}\Big) \tag{A.9}$$

E_{σ^2} is defined by (A.2), d_0 is a constant such that $dd_0 < 1$, $\hat{f}_{n,h_n}$ is the kernel density estimator of f defined by (2.3) with the kernel bandwidth h_n satisfying $h_n \to 0, \frac{\log h_n^{-1}}{nh_n^d} \to 0$, $\mathcal{V}_l$ is the Voronoi cell associated with $\mathbf{x}_l$, c_0 is a constant, $\theta_{lm} = \mathbb{1}_{\{\mathbf{y}_l \neq \mathbf{y}_m\}}$ is a class indicator function such that $\theta_{lm} = 1$ if $\mathbf{x}_l$ and $\mathbf{x}_m$ belongs to different classes, and 0 otherwise. Moreover, the equality in (A.8) holds when $\eta^{(i)} \equiv \frac{1}{Q}$ for $1 \le i \le Q$.

Lemma. *(Consistency of Kernel Density Estimator) Let the kernel bandwidth h_n of the Gaussian kernel K be chosen such that $h_n \to 0, \frac{\log h_n^{-1}}{nh_n^d} \to 0$. For any $P_X \in \mathcal{P}_X$, there exists a n_0 which depends on σ_0 and VC characteristics of K, when $n > n_0$, with probability greater than $1 - Lh_n^{E_{\sigma_0^2}}$ over the data $\{\mathbf{x}_l\}$,*

$$\left\|\hat{f}_{n,h_n}(x) - f(x)\right\|_\infty = \mathcal{O}\Big(\sqrt{\frac{\log h_n^{-1}}{nh_n^d}} + h_n^\gamma\Big) \tag{A.10}$$

where $\hat{f}_{n,h_n}$ is the kernel density estimator of f. Furthermore, for any $P_{XY} \in \mathcal{P}_{XY}$, when $n > n_0$, then with probability greater than $1 - 2Lh_n^{E_{\sigma_0^2}}$ over the data $\{\mathbf{x}_l\}$,

$$\left\|\hat{\eta}^{(i)}_{n,h_n}(x) - \eta^{(i)}(x)\right\|_\infty = \mathcal{O}\Big(\sqrt{\frac{\log h_n^{-1}}{nh_n^d}} + h_n^\gamma\Big) \tag{A.11}$$

for each $1 \le i \le Q$.

Lemma. *(Consistency of the Generalized Kernel Density Estimator) Suppose f is the probabilistic density function of $P_X \in \mathcal{P}_X$, and $f \le f_{\max}$. Let g be a bounded function defined on $\mathcal{X}$ and $g \in \Sigma_{\gamma,g_0}$, $0 < g_{\min} \le g \le g_{\max}$, and $e = \frac{f}{g}$. Define the generalized kernel density estimator of e as*

$$\hat{e}_{n,h} \triangleq \frac{1}{n}\sum_{l=1}^{n}\frac{K_h(x - \mathbf{x}_l)}{g(\mathbf{x}_l)} \tag{A.12}$$

Let $\sigma_g^2 = \frac{\|K\|_2^2 f_{\max}}{g_{\min}^2}$. *There exists* n_g *which depends on* σ_g *and the VC characteristics of* K *such that when* $n > n_g$, *with probability greater than* $1 - Lh_n^{E_{\sigma_g^2}}$ *over the data* $\{\mathbf{x}_l\}$,

$$\|\hat{e}_{n,h_n}(x) - e(x)\|_\infty = \mathcal{O}\Big(\sqrt{\frac{\log h_n^{-1}}{nh_n^d}} + h_n^\gamma\Big) \tag{A.13}$$

where h_n *is chosen such that* $h_n \to 0$, $\frac{\log h_n^{-1}}{nh_n^d} \to 0$.

Proof of Lemma 2

Proof. Since f satisfies assumption (A), applying Corollary 2.2 in [23], for $L, C > 0$ that depend solely on the VC characteristics of K and any $\lambda > C$, when $n > n_0$,

$$\Pr\left[\left\|\hat{f}_{n,h_n} - \mathbb{E}\left[\hat{f}_{n,h_n}\right]\right\|_\infty \geq \tau_n\right] \leq L\exp\left(-\frac{1}{L}\frac{\log(1+\lambda/4L)}{\lambda}\frac{nh_n^d\tau_n^2}{\sigma_0^2}\right) \tag{A.14}$$

where $\tau_n = \tau\sqrt{\frac{\log h_n^{-1}}{nh_n^d}}$, $\tau > \max\{1, C\sigma_0\sqrt{d/2+1}\}$.

Also,

$$\begin{aligned}
&\left\|\mathbb{E}\left[\hat{f}_{n,h_n}\right] - f(x)\right\|_\infty = \|\mathbb{E}_Z[K_h(x-Z)] - f(x)\|_\infty \\
&= \left\|\int_{\mathcal{X}} f(x - h_n z)K(z)\,dz - f(x)\int_{\mathcal{X}} K(z)\,dz\right\|_\infty \\
&\leq \int_{\mathcal{X}} \|f(x-h_nz) - f(x)\|_\infty K(z)\,dz \\
&\leq ch_n^\gamma \int_{\mathcal{X}} \|z\|^\gamma K(z)\,dx = ch_n^\gamma K_\gamma
\end{aligned} \tag{A.15}$$

because f is a Hölder-γ smooth function with Hölder constant $c = \sum_i \pi^{(i)} c_i$, and $\mathcal{X} = \mathbb{R}^d$. Based on (A.14) and (A.15), with probability greater than $1 - Lh_n^{E_{\sigma_0^2}}$ (since $h_n^{\tau^2 E_{\sigma_0^2}} < h_n^{E_{\sigma_0^2}}$ when $h_n < 1$ for sufficiently large n) over the data $\{\mathbf{x}_l\}$, (A.10) holds.

Moreover,

$$\begin{aligned}
&\left\|\hat{\eta}_{n,h_n}^{(i)}(x)-\eta^{(i)}(x)\right\|_\infty \\
&\leq \left\|\hat{\eta}_{n,h_n}^{(i)}(x)-\frac{\sum_{l=1}^{n}K_{h_n}(x-\mathbf{x}_l)\mathbb{1}_{\{\mathbf{y}_l=i\}}}{nf(x)}\right\|_\infty+\left\|\frac{\sum_{l=1}^{n}K_{h_n}(x-\mathbf{x}_l)\mathbb{1}_{\{\mathbf{y}_l=i\}}}{nf(x)}-\eta^{(i)}(x)\right\|_\infty \\
&\leq \left\|\frac{\sum_{l=1}^{n}K_{h_n}(x-\mathbf{x}_l)\mathbb{1}_{\{\mathbf{y}_l=i\}}}{n}\frac{f(x)-\hat{f}_{n,h_n}(x)}{f(x)\hat{f}_{n,h_n}(x)}\right\|_\infty+ \\
&\left\|\frac{1}{f(x)}\Big(\frac{\sum_{l=1}^{n}K_{h_n}(x-\mathbf{x}_l)\mathbb{1}_{\{\mathbf{y}_l=i\}}}{n}-\pi^{(i)}f^{(i)}(x)\Big)\right\|_\infty \\
&\leq \frac{1}{f_{\min}}\left\|f(x)-\hat{f}_{n,h_n}(x)\right\|_\infty+\frac{1}{f_{\min}}\left\|\frac{\sum_{l=1}^{n}K_{h_n}(x-\mathbf{x}_l)\mathbb{1}_{\{\mathbf{y}_l=i\}}}{n}-\pi^{(i)}f^{(i)}(x)\right\|_\infty
\end{aligned}$$

Similar to the proof of (A.10), with probability greater than $1-Lh_n^{E_{\sigma_0^2}}$ over the data $\{\mathbf{x}_l\}$, $\left\|\frac{\sum_{l=1}^{n}K_{h_n}(x-\mathbf{x}_l)\mathbb{1}_{\{\mathbf{y}_l=i\}}}{n}-\pi^{(i)}f^{(i)}(x)\right\|_\infty=\mathcal{O}\left(\sqrt{\frac{\log h_n^{-1}}{nh_n^d}}+h_n^\gamma\right)$. Also, with probability greater than $1-Lh_n^{E_{\sigma_0^2}}$, $\left\|\hat{f}_{n,h_n}(x)-f(x)\right\|_\infty=\mathcal{O}\left(\sqrt{\frac{\log h_n^{-1}}{nh_n^d}}+h_n^\gamma\right)$. Therefore, with probability greater than $1-2Lh_n^{E_{\sigma_0^2}}$, (A.11) holds. □

Note that when $\sum_n h_n^{\tau^2 E_{\sigma_0^2}}<\infty$, with probability 1,

$$\overline{\lim_{n\to\infty}}\left\|\hat{f}_{n,h_n}(x)-f(x)\right\|_\infty=\mathcal{O}\Big(\sqrt{\frac{\log h_n^{-1}}{nh_n^d}}+h_n^\gamma\Big) \tag{A.16}$$

and

$$\overline{\lim_{n\to\infty}}\left\|\hat{\eta}_{n,h_n}^{(i)}(x)-\eta^{(i)}(x)\right\|_\infty=\mathcal{O}\Big(\sqrt{\frac{\log h_n^{-1}}{nh_n^d}}+h_n^\gamma\Big) \tag{A.17}$$

for each $1\leq i\leq Q$, which follow from the Borel-Cantelli lemma.

Proof of Lemma 3

Proof. We consider the class of functions

$$\mathcal{F} \triangleq \{K\left(\frac{t-\cdot}{h}\right), t \in \mathbb{R}^d, h \neq 0\} \quad \mathcal{F}_g \triangleq \{\frac{K\left(\frac{t-\cdot}{h}\right)}{g\left(\cdot\right)}, t \in \mathbb{R}^d, h \neq 0\}$$

Since $\mathcal{F}$ is a bounded VC class of measurable functions, there exist positive numbers A and v such that for every probability measure P on $\mathbb{R}^d$ for which $\int F^2 dP < \infty$ and any $0 < \tau < 1$,

$$N\left(\mathcal{F}, \|\cdot\|_{L_2(P)}, \tau \|F\|_{L_2(P)}\right) \leq \left(\frac{A}{\tau}\right)^v \tag{A.18}$$

For any $t_1, t_2 \in \mathbb{R}^d$ and $h_1, h_2 > 0$,

$$\left\| \frac{K\left(\frac{t_1-\cdot}{h_1}\right)}{g\left(\cdot\right)} - \frac{K\left(\frac{t_2-\cdot}{h_2}\right)}{g\left(\cdot\right)} \right\|_{L_2(P)} \leq \frac{1}{g_{\min}} \left\| K\left(\frac{t_1-\cdot}{h_1}\right) - K\left(\frac{t_2-\cdot}{h_2}\right) \right\|_{L_2(P)}$$

Let $B_{\mathcal{F}}\left(t_0, h_0, \delta\right) \triangleq \{(t,h) : \left\| K\left(\frac{t-\cdot}{h}\right) - K\left(\frac{t_0-\cdot}{h_0}\right) \right\|_{L_2(P)} \leq \delta, h \neq 0\}$, and $B_{\mathcal{F}_g}\left(t_0, h_0, \delta\right) \triangleq \{(t,h) : \left\| \frac{K\left(\frac{t-\cdot}{h}\right)}{g(\cdot)} - \frac{K\left(\frac{t_0-\cdot}{h_0}\right)}{g(\cdot)} \right\|_{L_2(P)} \leq \delta\}$. Then $B_{\mathcal{F}}\left(t_0, h_0, \delta\right) \subseteq B_{\mathcal{F}_g}\left(t_0, h_0, \frac{\delta}{g_{\min}}\right)$.

We choose the envelope function for $\mathcal{F}_g$ as $F_g = \frac{F}{g_{\min}}$ and $|u_g| \leq F_g$ for any $u_g \in \mathcal{F}_g$. There is a bijection between $\mathcal{F}$ and $\mathcal{F}_g$, so

$$N\left(\mathcal{F}_g, \|\cdot\|_{L_2(P)}, \tau \|F_g\|_{L_2(P)}\right) \leq \left(\frac{A}{\tau}\right)^v \tag{A.19}$$

So that $\mathcal{F}_g$ is also a bounded VC class. The conclusion (A.13) follows from an argument similar to that in the proof of Theorem 1 and Corollary 2.2 in [23]. □

Similarly, when $\sum_n h_n^{\tau^2 E_{\sigma_g^2}} < \infty$ (here $\tau > \max\{1, C\sigma_g\sqrt{d/2+1}\}$), with probability 1,

$$\overline{\lim_{n\to\infty}} \left\|\hat{e}_{n,h_n}\left(x\right) - e\left(x\right)\right\|_\infty = \mathcal{O}\left(\sqrt{\frac{\log h_n^{-1}}{n h_n^d}} + h_n^\gamma\right)$$

Proof of Lemma 1

Proof. Let $P_{XY} \in \mathcal{P}_{XY}$. It can be verified that

$$R\left(\mathrm{PI}_S\right) = \sum_{i,j=1,\ldots,Q,i\neq j} \mathbb{E}_X\left[\eta^{(i)}\left(X\right)\Pr\left[\mathrm{PI}_S\left(X\right)=j\right]\right] \tag{A.20}$$

According to Lemma 2 and (A.20), with probability greater than $1-2QLh_n^{E_{\sigma_0^2}}$,

$$R\left(\mathrm{PI}_S\right) = \sum_{i\neq j} \mathbb{E}_X\left[\hat{\eta}^{(i)}_{n,h_n}\left(X\right)\Pr\left[\mathrm{PI}_S\left(X\right)=j\right]\right] + \mathcal{O}\left(\sqrt{\frac{\log h_n^{-1}}{nh_n^d}} + h_n^{\gamma}\right)$$

Denote by $\{\mathbf{R}_1, \mathbf{R}_2, \ldots \mathbf{R}_Q\}$ the decision regions of PI_S, then $\hat{\eta}^{(i)}_{n,h_n} \geq \hat{\eta}^{(i')}_{n,h_n}$ for all $i' \neq i$ on each $\mathbf{R}_i$, and

$$\begin{aligned}
&\sum_{i,j=1,\ldots,Q,i\neq j} \mathbb{E}_X\left[\hat{\eta}^{(i)}_{n,h_n}\left(X\right)\Pr\left[\mathrm{PI}_S\left(X\right)=j\right]\right] \\
&= \sum_{i,j=1,\ldots,Q,i\neq j} \mathbb{E}_{X\in\mathbf{R}_j}\left[\hat{\eta}^{(i)}_{n,h_n}\left(X\right)\cdot\sum_{k=1}^{Q}\hat{\eta}^{(k)}_{n,h_n}\left(X\right)\right] \\
&= \mathbb{E}_X\left[\left(\sum_{k=1}^{Q}\hat{\eta}^{(k)}_{n,h_n}\left(X\right)\right)^2\right] - \sum_{i=1}^{Q}\mathbb{E}_{X\in\mathbf{R}_i}\left[\hat{\eta}^{(i)}_{n,h_n}\left(X\right)\cdot\sum_{k=1}^{Q}\hat{\eta}^{(k)}_{n,h_n}\left(X\right)\right] \\
&\leq \mathbb{E}_X\left[\left(\sum_{k=1}^{Q}\hat{\eta}^{(k)}_{n,h_n}\left(X\right)\right)^2\right] - \sum_{i=1}^{Q}\mathbb{E}_X\left[\left(\hat{\eta}^{(i)}_{n,h_n}\left(X\right)\right)^2\right] \\
&= \sum_{i,j=1,\ldots,Q,i\neq j} \mathbb{E}_X\left[\hat{\eta}^{(i)}_{n,h_n}\left(X\right)\hat{\eta}^{(j)}_{n,h_n}\left(X\right)\right]
\end{aligned} \tag{A.21}$$

Therefore we obtain (A.3), and the equality in (A.3) holds when $\hat{\eta}^{(i)}_{n,h_n} \equiv \frac{1}{Q}$ for $1 \leq i \leq Q$. □

Proof of Theorem 1

Proof. By Lemma 1 and 2, there exists an $n^{(1)}$ which depends on σ_0 and the VC characteristics of K, when $n > n^{(1)}$, with probability greater than $1-2QLh_n^{E_{\sigma_0^2}}$,

$$R_n^{\mathrm{PI}} = \sum_{i\neq j} \mathbb{E}_X\left[\hat{\eta}^{(i)}_{n,h_n}\left(X\right)\hat{\eta}^{(j)}_{n,h_n}\left(X\right)\right]$$

$$= \sum_{i \neq j} \mathbb{E}_X \left[\eta^{(i)}(X) \eta^{(j)}(X) \right] + \mathcal{O}\left(\sqrt{\frac{\log h_n^{-1}}{n h_n^d}} + h_n^\gamma \right) \tag{A.22}$$

Note that

$$\mathbb{E}_X \left[\eta^{(i)}(X) \eta^{(j)}(X) \right] = \int_{\mathcal{X}} \frac{\pi^{(i)} f^{(i)}(x)}{f^{\frac{1}{2}}(x)} \cdot \frac{\pi^{(j)} f^{(j)}(x)}{f^{\frac{1}{2}}(x)} \, \mathrm{d}x,$$

Using the generalized kernel density estimator (A.12), we obtain the kernel estimator $\tilde{\eta}_{n,h_n}^{(i)}$ of $\frac{\pi^{(i)} f^{(i)}(x)}{f^{\frac{1}{2}}(x)}$ as below:

$$\tilde{\eta}_{n,h_n}^{(i)}(x) = \frac{1}{n} \sum_{l=1}^{n} \frac{K_{h_n}(x - \mathbf{x}_l) \mathbb{I}_{\{\mathbf{y}_l = i\}}}{f^{\frac{1}{2}}(\mathbf{x}_l)} \tag{A.23}$$

By Lemma 3, there exists an $n^{(2)}$ which depends on σ_1 and the VC characteristics of K, when $n > n^{(2)}$, with probability greater than $1 - QLh_n^{E_{\sigma_1^2}}$,

$$\sum_{i \neq j} \mathbb{E}_X \left[\eta^{(i)}(X) \eta^{(j)}(X) \right] = \sum_{i \neq j} \mathbb{E}_X \left[\tilde{\eta}_{n,h_n}^{(i)}(X) \tilde{\eta}_{n,h_n}^{(j)}(X) \right] + \mathcal{O}\left(\sqrt{\frac{\log h_n^{-1}}{n h_n^d}} + h_n^\gamma \right) \tag{A.24}$$

By convolution theorem of Gaussian kernels,

$$\sum_{i \neq j} \mathbb{E}_X \left[\tilde{\eta}_{n,h_n}^{(i)}(X) \tilde{\eta}_{n,h_n}^{(j)}(X) \right] = \frac{1}{n^2} \sum_{l,m} \frac{K_{\sqrt{2}h_n}(\mathbf{x}_l - \mathbf{x}_m)}{f^{\frac{1}{2}}(\mathbf{x}_l) f^{\frac{1}{2}}(\mathbf{x}_m)} \theta_{lm}$$

Letting $\tilde{h}_n = \sqrt{2} h_n$, there exists $n^{(3)}$ depending on σ_0 and the VC characteristics of K, when $n > n^{(3)}$, with probability greater than $1 - L\tilde{h}_n^{E_{\sigma_0^2}}$, $\| \hat{f}_{n,\tilde{h}_n}(x) - f(x) \|_\infty = \mathcal{O}\left(\sqrt{\frac{\log \tilde{h}_n^{-1}}{n \tilde{h}_n^d}} + \tilde{h}_n^\gamma \right)$ and $\| \hat{f}_{n,\tilde{h}_n}(x) - f(x) \|_\infty \leq \frac{f_{\min}}{2}$. It follows that $\sup_{x \in \mathbb{R}^d} \hat{f}_{n,\tilde{h}_n}(x) \leq f_{\max} + \frac{f_{\min}}{2}$, $\inf_{x \in \mathbb{R}^d} \hat{f}_{n,\tilde{h}_n}(x) \geq \frac{f_{\min}}{2}$, and

$$\left| \sum_{i \neq j} \mathbb{E}_X \left[\tilde{\eta}_{n,h_n}^{(i)}(X) \tilde{\eta}_{n,h_n}^{(j)}(X) \right] - \frac{1}{n^2} \sum_{l,m} G_{lm,\tilde{h}_n} \theta_{lm} \right|$$

$$
\begin{aligned}
&\leq \frac{1}{n^2}\sum_{l,m} K_{\tilde{h}_n}(\mathbf{x}_l - \mathbf{x}_m)\frac{\left|f^{\frac{1}{2}}(\mathbf{x}_l)f^{\frac{1}{2}}(\mathbf{x}_m) - \hat{f}^{\frac{1}{2}}_{n,\tilde{h}_n}(\mathbf{x}_l)\hat{f}^{\frac{1}{2}}_{n,\tilde{h}_n}(\mathbf{x}_m)\right|}{\hat{f}^{\frac{1}{2}}_{n,\tilde{h}_n}(\mathbf{x}_l)\hat{f}^{\frac{1}{2}}_{n,\tilde{h}_n}(\mathbf{x}_m)f^{\frac{1}{2}}(\mathbf{x}_l)f^{\frac{1}{2}}(\mathbf{x}_m)} \\
&= \mathcal{O}\left(\sqrt{\frac{\log \tilde{h}_n^{-1}}{n\tilde{h}_n^d}} + \tilde{h}_n^{\gamma}\right)\cdot\frac{1}{n}\sum_{l=1}^{n}\hat{f}_{n,\tilde{h}_n}(\mathbf{x}_l) \\
&= \mathcal{O}\left(\sqrt{\frac{\log \tilde{h}_n^{-1}}{n\tilde{h}_n^d}} + \tilde{h}_n^{\gamma}\right) = \mathcal{O}\left(\sqrt{\frac{\log {h_n}^{-1}}{nh_n^d}} + h_n^{\gamma}\right) \qquad \text{(A.25)}
\end{aligned}
$$

since $\tilde{h}_n = \sqrt{2}h_n$. Taking $n_1 = \max\{n^{(1)}, n^{(2)}, n^{(3)}\}$, it follows from (A.22), (A.24) and (A.25) that with probability greater than $1-2QLh_n^{E_{\sigma_0^2}} - L(\sqrt{2}h_n)^{E_{\sigma_0^2}} - QLh_n^{E_{\sigma_1^2}}$,

$$
R_n^{\mathrm{PI}} = \frac{1}{n^2}\sum_{l,m} G_{lm,\sqrt{2}h_n}\theta_{lm} + \mathcal{O}\left(\sqrt{\frac{\log h_n^{-1}}{nh_n^d}} + {h_n}^{\gamma}\right) \qquad \text{(A.26)}
$$

and (A.5) is verified by (A.26). □

Proof of Corollary 1

Suppose the data $\{\mathbf{x}_i\}_{i=1}^n$ lies on a domain $\Omega \subseteq R^d$. Let f be the probability density function on Ω, S be the cluster boundary which separates Ω into two parts S_1 and S_2 (see Figure A.1). Let the domain of f be restricted to Ω in assumption (A) and (B). Based on the analysis in the beginning of this document, Theorem $1-4$ and Lemma $1-2$ still hold and the proofs remain almost unchanged.

The Low Density Separation assumption favors the cluster boundary with low volume, i.e. $\int_S f(s)ds$. Corollary 1 reveals the relationship between the error of the plug-in classifier and the weighted volume of the cluster boundary.

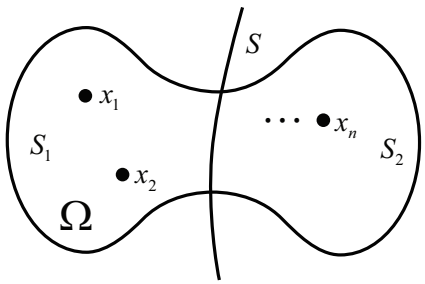


Figure A.1: Illustration of the hyperplane S for Low Density Separation.

Proof. Firstly, we show that when restricting the support of the marginal distribution P_X to a subset $\Omega \subset \mathbb{R}^d$ which is not necessarily full-dimensional, Theorem 1 and lemma 1-3 still hold and our derived bounds are still valid. To see this, we only need to show that the following class of functions $\mathcal{F}_\Omega$ is a bounded VC class of measurable functions.

$$\mathcal{F}_\Omega \triangleq \{K\left(\frac{t-\cdot}{h}\right), t \in \Omega, h \neq 0\} \tag{A.27}$$

Since we already know that the class of functions $\mathcal{F}$ defined below (also in the chapter) is a bounded VC class of measurable functions with respect to the envelope function F,

$$\mathcal{F} \triangleq \{K\left(\frac{t-\cdot}{h}\right), t \in \mathbb{R}^d, h \neq 0\} \tag{A.28}$$

we have $N\left(\mathcal{F}, \|\cdot\|_{L_2(P)}, \tau \|F\|_{L_2(P)}\right) \leq \left(\frac{A}{\tau}\right)^v$ for every probability measure P on $\mathbb{R}^d$ for which $\int F^2 dP < \infty$ and any $0 < \tau < 1$. $N\left(\mathcal{T}, \hat{d}, \epsilon\right)$ is defined as the minimal number of open $\hat{d}$-balls of radius ϵ required to cover $\mathcal{T}$ in the metric space $\left(\mathcal{T}, \hat{d}\right)$. Letting $\{B_i\}$ be the $N\left(\mathcal{F}, \|\cdot\|_{L_2(P)}, \tau \|F\|_{L_2(P)}\right)$ open balls which cover $\mathcal{F}$, then $\{B_i \cap \mathcal{F}_\Omega\}$ is the set of balls which cover $\mathcal{F}_\Omega$ since $\mathcal{F}_\Omega \subset \mathcal{F}$. It follows that $\mathcal{F}_\Omega$ is also a bounded VC class of measurable functions with respect to the envelope function F.

According to Theorem 3 in [1], for any $\varepsilon \in \left(0, \frac{1}{2}\right)$, there exists constant C such that for all h satisfying $0 < h < \sqrt{\tau}(2d)^{-\frac{e}{2(e-1)}}$,

$$\left| \frac{\sqrt{\pi}}{h} \int_{S_2} \int_{S_1} K_{\sqrt{2}h}(x-y) \psi_{\sqrt{2}h}(x) \psi_{\sqrt{2}h}(y) dx dy - \int_S f(s) ds \right| < Ch^{2\varepsilon} \tag{A.29}$$

where $\psi_h(x) = \frac{f(x)}{\sqrt{\int_\Omega K_h(x-z) f(z) dz}}$, τ is the radius of the largest ball that can be placed tangent to the manifold Ω.

Let $\tau_n = C_0$ for some constant $C_0 < \frac{f_{\min}}{2}$ in equation (A.14) in the proof of Lemma 2. Then there exists n_0 depending on σ_0 and the VC characteristics of K, when $n > n_0$, with probability greater than $1 - L\exp\left(-\frac{1}{L}\frac{\log(1+\lambda/4L)}{\lambda}\frac{n(\sqrt{2}h_n)^d C_0^2}{\sigma_0^2}\right)$, $\|\hat{f}_{n,\sqrt{2}h_n}(x) - f(x)\|_\infty \leq \frac{f_{\min}}{2}$. Denote by A the event that $\|\hat{f}_{n,\sqrt{2}h_n}(x) - f(x)\|_\infty \leq \frac{f_{\min}}{2}$.

Define

$$R\big((\mathbf{x}_1,\mathbf{y}_1),\ldots,(\mathbf{x}_n,\mathbf{y}_n)\big)=\frac{\sqrt{\pi}}{2h_n}\hat{R}_n\left(\mathrm{PI}_S\right)$$
$$=\frac{1}{n^2}\frac{\sqrt{\pi}}{2h_n}\sum_{l,m}\frac{K_{\sqrt{2}h_n}\left(\mathbf{x}_l-\mathbf{x}_m\right)}{\hat{f}^{\frac{1}{2}}_{n,\sqrt{2}h_n}\left(\mathbf{x}_l\right)\hat{f}^{\frac{1}{2}}_{n,\sqrt{2}h_n}\left(\mathbf{x}_m\right)}\theta_{lm}$$

with $\theta_{lm}=\mathbb{1}_{\{\mathbf{y}_l\neq\mathbf{y}_m\}}$, then the bounded difference is verified when A holds:

$$\begin{aligned}
&\Big|R\big((\mathbf{x}_1,\mathbf{y}_1),\ldots,(\mathbf{x}_l,\mathbf{y}_l),\ldots,(\mathbf{x}_n,\mathbf{y}_n)\big)-R\big((\mathbf{x}_1,\mathbf{y}_1),\ldots,(\mathbf{x}_l^{'},\mathbf{y}_l^{'}),\ldots,(\mathbf{x}_n,\mathbf{y}_n)\big)\Big|\\
&\leq\frac{1}{n^2}\frac{\sqrt{\pi}}{h_n}\sum_{m}\left(\frac{K_{\sqrt{2}h_n}(\mathbf{x}_l-\mathbf{x}_m)}{\hat{f}^{\frac{1}{2}}_{n,\sqrt{2}h_n}\left(\mathbf{x}_l\right)\hat{f}^{\frac{1}{2}}_{n,\sqrt{2}h_n}\left(\mathbf{x}_m\right)}+\frac{K_{\sqrt{2}h_n}(\mathbf{x}_l^{'}-\mathbf{x}_m)}{\hat{f}^{\frac{1}{2}}_{n,\sqrt{2}h_n}\left(\mathbf{x}_l^{'}\right)\hat{f}^{\frac{1}{2}}_{n,\sqrt{2}h_n}\left(\mathbf{x}_m\right)}\right)\\
&=\frac{1}{n^2}\frac{\sqrt{\pi}}{h_n}\sum_{m}\frac{\hat{f}^{\frac{1}{2}}_{n,\sqrt{2}h_n}(\mathbf{x}_l^{'})K_{\sqrt{2}h_n}(\mathbf{x}_l-\mathbf{x}_m)+\hat{f}^{\frac{1}{2}}_{n,\sqrt{2}h_n}(\mathbf{x}_l)K_{\sqrt{2}h_n}(\mathbf{x}_l^{'}-\mathbf{x}_m)}{\hat{f}^{\frac{1}{2}}_{n,\sqrt{2}h_n}(\mathbf{x}_l^{'})\hat{f}^{\frac{1}{2}}_{n,\sqrt{2}h_n}(\mathbf{x}_l)\hat{f}^{\frac{1}{2}}_{n,\sqrt{2}h_n}\left(\mathbf{x}_m\right)}\\
&\leq\frac{C_1}{nh_n^{d+1}}
\end{aligned}\tag{A.30}$$

where C_1 a constant determined by $f_{\min},f_{\max},d$. According to McDiarmid's inequality,

$$\begin{aligned}
&\Pr\left[\left|R\big((\mathbf{x}_1,\mathbf{y}_1),\ldots,(\mathbf{x}_n,\mathbf{y}_n)\big)-\mathbb{E}R\big((\mathbf{x}_1,\mathbf{y}_1),\ldots,(\mathbf{x}_n,\mathbf{y}_n)\big)\right|\geq\varepsilon_1\mid A\right]\\
&\leq 2\exp\left(-\frac{2nh_n^{2d+2}\varepsilon_1^2}{C_1^2}\right)
\end{aligned}\tag{A.31}$$

Also, the expectation of $R\big((\mathbf{x}_1,\mathbf{y}_1),\ldots,(\mathbf{x}_n,\mathbf{y}_n)\big)$ satisfies

$$\begin{aligned}
&\mathbb{E}R\big((\mathbf{x}_1,\mathbf{y}_1),\ldots,(\mathbf{x}_n,\mathbf{y}_n)\big)\\
&=\frac{\sqrt{\pi}}{2h_n}\int_{S_2}\int_{S_1}\frac{K_{\sqrt{2}h_n}\left(x-y\right)}{\hat{f}^{\frac{1}{2}}_{n,\sqrt{2}h_n}\left(x\right)\hat{f}^{\frac{1}{2}}_{n,\sqrt{2}h_n}\left(y\right)}f\left(x\right)f\left(y\right)dxdy\\
&+\frac{\sqrt{\pi}}{2h_n}\int_{S_1}\int_{S_2}\frac{K_{\sqrt{2}h_n}\left(x-y\right)}{\hat{f}^{\frac{1}{2}}_{n,\sqrt{2}h_n}\left(x\right)\hat{f}^{\frac{1}{2}}_{n,\sqrt{2}h_n}\left(y\right)}f\left(x\right)f\left(y\right)dxdy\\
&=\frac{\sqrt{\pi}}{h_n}\int_{S_2}\int_{S_1}\frac{K_{\sqrt{2}h_n}\left(x-y\right)}{\hat{f}^{\frac{1}{2}}_{n,\sqrt{2}h_n}\left(x\right)\hat{f}^{\frac{1}{2}}_{n,\sqrt{2}h_n}\left(y\right)}f\left(x\right)f\left(y\right)dxdy
\end{aligned}$$

Moreover, the square of the denominator of ψ_h is the expectation of $\hat{f}_{n,h_n}$, i.e.

$\int_{\Omega} K_h(x-z) f(z) dz = \mathbb{E}\left[\hat{f}_{n,h}\right]$. Again, by equation (A.14) in the proof of Lemma 2,

$$\begin{aligned}
&\Pr\Big[\Big|\int_{S_2}\int_{S_1} \frac{K_{\sqrt{2}h_n}(x-y)}{\hat{f}^{\frac{1}{2}}_{n,\sqrt{2}h_n}(x)\,\hat{f}^{\frac{1}{2}}_{n,\sqrt{2}h_n}(y)} f(x) f(y) dxdy - \\
&\int_{S_2}\int_{S_1} K_{\sqrt{2}h_n}(x-y)\,\psi_{\sqrt{2}h_n}(x)\,\psi_{\sqrt{2}h_n}(y) dxdy\Big| \geq \varepsilon_2 \mid A\Big] \\
&\leq \Pr\left[\left\|\mathbb{E}\left[\hat{f}_{n,\sqrt{2}h_n}(x)\right] - \hat{f}_{n,\sqrt{2}h_n}(x)\right\|_{\infty} \geq C_2\varepsilon_2\right] \\
&\leq L\exp\Big(-\frac{1}{L}\frac{\log(1+\lambda/4L)}{\lambda}\frac{n(\sqrt{2}h_n)^d C_2^2\varepsilon_2^2}{\sigma_0^2}\Big)
\end{aligned} \tag{A.32}$$

where C_2 is a constant. Note that $\Pr[A] \geq 1 - L\exp\left(-\frac{1}{L}\frac{\log(1+\lambda/4L)}{\lambda}\frac{n(\sqrt{2}h_n)^d C_0^2}{\sigma_0^2}\right)$, by (A.29), (A.31) and (A.32) and the application of the Borel-Cantelli lemma, (A.7) is verified. □

Proof of Theorem 2

Proof. Denote the support of P_X by $\mathcal{X}$. Since $\mathcal{X}$ is bounded in $\mathbb{R}^d$, we construct the τ-cover of $\mathcal{X}$ which is a sequence of sets $\{\Omega_1, \Omega_2, ..., \Omega_{\mathcal{R}}\}$ such that $\mathcal{X} \subseteq \bigcup_{r=1}^{\mathcal{R}} \Omega_r$ and each Ω_r is a box of length τ in $\mathbb{R}^d$, $1 \leq r \leq \mathcal{R}$, $\mathcal{R} = \left(\frac{2M_0}{\tau}\right)^d$. Let $A = \bigcap_{r=1}^{\mathcal{R}} \{\Omega_r \bigcap \{\mathbf{x}_l\}_{l=1}^n \neq \emptyset\}$ indicate the event that each Ω_r contains at least one data point from $\{\mathbf{x}_l\}_{l=1}^n$, then,

$$\begin{aligned}
&\Pr[A] \geq 1 - \mathcal{R}(1 - \Pr[\Omega_1])^n = 1 - \mathcal{R}e^{n\log(1-\Pr[\Omega_1])} \\
&\geq 1 - \mathcal{R}e^{-n\Pr[\Omega_1]} \geq 1 - (\frac{2M_0}{\tau})^d e^{-nf_{\min}\tau^d}
\end{aligned}$$

So A holds with probability greater than $1 - (\frac{2M_0}{\tau})^d e^{-nf_{\min}\tau^d}$. Denote by $\tilde{X}$ the nearest neighbor of X among $\{\mathbf{x}_l\}_{l=1}^n$, and $\tilde{Y}$ is the label of $\tilde{X}$. Note that $\left\|X - \tilde{X}\right\|_2 \leq \sqrt{d}\tau$ if $X \in \Omega_r$ for each r. For any $P_{XY} \in \mathcal{P}_{XY}$, some calculation shows that $\exists \tilde{c}_i > 0, \left|\eta^{(i)}(x) - \eta^{(i)}(y)\right| \leq \tilde{c}_i \|x-y\|^{\gamma}$, so that $\eta^{(i)}$ is also Hölder-γ smooth with Hölder constant $\tilde{c}_i$. We then have

$$R(\mathrm{NN}_S) = \mathbb{E}_{(X,Y)}\left[Y \neq \tilde{Y}\right] \tag{A.33}$$

$$
\begin{aligned}
&= \sum_{r=1}^{\mathcal{R}} \mathbb{E}_X \left[\left(1 - \eta^{(\tilde{Y})}(X) \right) \mathbb{1}_{\{X \in \Omega_r\}} \right] \\
&\leq \sum_{r=1}^{\mathcal{R}} \mathbb{E}_X \left[\left(1 - \eta^{(\tilde{Y})}\left(\tilde{X}\right) + \tilde{c}_{\tilde{Y}} \left(\sqrt{d}\tau\right)^{\gamma} \right) \mathbb{1}_{\{X \in \Omega_r\}} \right] \\
&\leq \sum_{r=1}^{\mathcal{R}} \mathbb{E}_X \left[\left(1 - \eta^{(\tilde{Y})}\left(\tilde{X}\right) \right) \mathbb{1}_{\{X \in \Omega_r\}} \right] + \underbrace{\max_i \tilde{c}_i}_{\triangleq c_0} \left(\sqrt{d}\tau\right)^{\gamma}
\end{aligned}
$$

Let $N_r = \{\mathbf{x}_s \in \{\mathbf{x}_l\}_{l=1}^n \mid \mathbf{x}_s = \tilde{X} \text{ for some } \mathrm{X} \in \Omega_\mathrm{r}\}$ wherein each element is the nearest neighbor of some $X \in \Omega_r$, and $\Omega_{rs} = \{X \in \Omega_r \mid \tilde{X} = \mathbf{x}_s, \mathbf{x}_s \in N_r\}$ which is a subregion of Ω_r such that all $X \in \Omega_{rs}$ takes $\mathbf{x}_s$ as its nearest neighbor. Then $\Omega_r = \bigcup_{s:\mathbf{x}_s \in N_r} \Omega_{rs}$, and $\tilde{X} = \mathbf{x}_s$ for $X \in \Omega_{rs}$. Since $\{\mathbf{x}_l\}_{l=1}^n \subset \bigcup_{r=1}^{\mathcal{R}} \Omega_r$, each $\mathbf{x}_l$ should be the nearest neighbor of some $X \in \Omega_r$, $1 \leq r \leq \mathcal{R}$, so that $\{\mathbf{x}_l\}_{l=1}^n = \bigcup_{r=1}^{\mathcal{R}} N_r$.

Based on Theorem 1, with probability greater than $1 - 2QLh_n^{E_{\sigma_0^2}}$,

$$
\begin{aligned}
&\sum_{r=1}^{\mathcal{R}} \mathbb{E}_X \left[\left(1 - \eta^{(\tilde{Y})}\left(\tilde{X}\right) \right) \mathbb{1}_{\{X \in \Omega_r\}} \right] \\
&= \sum_{s=1}^{n} \left[1 - \eta^{(y_s)}(\mathbf{x}_s) \right] \int_{\mathcal{V}_s} f(x)\, \mathrm{d}x \\
&= \sum_{s=1}^{n} \left\{ \left[1 - \hat{\eta}_{n,h_n}^{(y_s)}(\mathbf{x}_s) \right] \int_{\mathcal{V}_s} \hat{f}_{n,h_n}(x)\, \mathrm{d}x \right\} + \mathcal{O}\left(\sqrt{\frac{\log h_n^{-1}}{n h_n^d}} + h_n^{\gamma} \right) \\
&= \frac{1}{n} \sum_{l<m} H_{lm}\theta_{lm} + \mathcal{O}\left(\sqrt{\frac{\log h_n^{-1}}{n h_n^d}} + h_n^{\gamma} \right)
\end{aligned} \tag{A.34}
$$

where $\mathcal{V}_s$ is the Voronoi cell associated with $\mathbf{x}_s$, which is the set of points whose nearest neighbor is $\mathbf{x}_s$: $\mathcal{V}_s = \bigcap_{l:l\neq s} \{x \in \mathcal{X} \mid \|x - \mathbf{x}_s\|_2 \leq \|x - \mathbf{x}_l\|_2\}$. Combining (A.33) and (A.34),

$$
R(\mathrm{NN}_S) \leq \frac{1}{n} \sum_{l<m} H_{lm}\theta_{lm} + c_0 \left(\sqrt{d}\tau\right)^{\gamma} + \mathcal{O}\left(\sqrt{\frac{\log h_n^{-1}}{n h_n^d}} + h_n^{\gamma} \right) \tag{A.35}
$$

Moreover, the equality in (A.35) holds if the equality in (A.33) holds, e.g. $\eta^{(i)} \equiv \frac{1}{Q}$ for $1 \leq i \leq Q$. □

A.1.2 Algorithm and Experiments

The objective function of our pairwise clustering method PIEC is

$$\Psi(e) = \sum_{l=1}^{n} \exp\left(-G_{le_l,\sqrt{2}h_n}\right) + \lambda \sum_{l,m} \left(\tilde{\theta}_{lm} G_{lm,\sqrt{2}h_n} + \rho_{lm}(e_l, e_m)\right) \tag{A.36}$$

where ρ_{lm} is a function to enforce the consistency of the cluster indicators:

$$\rho_{lm}(e_l, e_m) = \begin{cases} \infty & e_m = l, e_l \neq l \text{ or } e_l = m, e_m \neq m \\ 0 & \text{otherwise} \end{cases}$$

The minimization of the objective function is converted to a MAP (Maximum a Posterior) problem in the pairwise MRF. (A.36) is minimized by Max-Product Belief Propagation (BP) in two steps:

Message Passing: BP iteratively passes messages along each edge according to

$$m_{lm}^{t}(e_m) = \min_{e_l} \left(M_{lm}^{t-1}(e_l) + \tilde{\theta}_{lm} G_{lm,\sqrt{2}h_n} + \rho_{lm}(e_l, e_m)\right) \tag{A.37}$$

$$M_{lm}^{t}(e_l) \triangleq \sum_{k \in \mathcal{N}(l)\backslash m} m_{kl}^{t}(e_l) + u_l(e_l) \tag{A.38}$$

where m_{lm}^{t} is the message sent from node l to node m in iteration t, $\mathcal{N}(l)$ is the set of neighbors of node l.

Inferring the optimal label: After the message passing converges or the maximal number of iterations is achieved, the final belief for each node is $b_l(e_l) = \sum_{k \in \mathcal{N}(l)} m_{kl}^{T}(e_l) + u_l(e_l)$, T is the number of iterations of message passing. The resultant optimal e_l^* is $e_l^* = \arg\min_{e_l} b_l(e_l)$.

AP (Affinity Propagation) controls the cluster numbers by a parameter called preference. We first estimate the lower bound and upper bound for the preference using the routine functions provided by the authors [32], then evenly sample 170 preference values between its upper bound and lower bound, and run AP with each sampled preference value. CEB (Convex Clustering with Exemplar-Based Model) produces different cluster numbers by varying the scale $\beta\beta_0$ which controls the shape of the mixture components. Likewise, we evenly sample 170 values between $[0.1, 2]$ for β, and $\beta_0 = n^2 \log n / \sum_{i,j} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$ according to [33]. Also,

we normalize the BT data set so that it has unit column variance, since the column variances of BT vary significantly (the largest column variance is 18580 while the smallest one is 0.0686).

A.2 Supplementary Document for Subspace Learning with ℓ^0-Graph

We present detailed proof of theorems and lemmas in Chapter 4, and provide additional experimental results in this supplementary.

A.2.1 Proof of Theorem 8

The ℓ^0-induced sparse subspace clustering solves the following problem:

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \quad s.t.\ \boldsymbol{X} = \boldsymbol{X}\boldsymbol{\alpha},\ \mathrm{diag}(\boldsymbol{\alpha}) = \mathbf{0} \tag{A.39}$$

Theorem 8. *(ℓ^0-Induced Almost Surely Subspace-Sparse Representation) Suppose the data $\boldsymbol{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ lie in a union of K distinct subspaces $\{\mathcal{S}_k\}_{k=1}^K$ of dimensions $\{d_k\}_{k=1}^K$, i.e. $\mathcal{S}_k \neq \mathcal{S}_{k'}$ for $k \neq k'$. Let $\boldsymbol{X}^{(k)} \in \mathbb{R}^{d\times n_k}$ denote the data that belong to subspace $\mathcal{S}_k$, and $\sum_{k=1}^K n_k = n$. When $n_k \geq d_k + 1$, if the data belonging to each subspace are generated i.i.d. from arbitrary unknown continuous distribution supported on that subspace,[1] then with probability 1, the optimal solution to (A.39), denoted by $\boldsymbol{\alpha}^*$, is a subspace-sparse representation, i.e. nonzero elements in $\boldsymbol{\alpha}^{*i}$ corresponds to the data that lie in the same subspace as $\mathbf{x}_i$.*

To prove Theorem 8, we need the claims below, which show that the probability that a point lies in a low dimensional subspace in any subspace $\mathcal{S}_k$ for $k = 1 \ldots K$ is 0, and any $L \leq d_k$ points in $\boldsymbol{X}^{(k)}$ are most surely linearly independent, under the assumptions of Theorem 8.

Claim 1. *Under the assumptions of Theorem 8, for a random data point $\mathbf{x} \in \mathcal{S}_k$ that is generated according to a continuous distribution supported on $\mathcal{S}_k$, the probability that $\mathbf{x}$ lies in a hyperplane H in $\mathcal{S}_k$ which has dimension less than d_k is zero, i.e. $\Pr[\mathbf{x} \in H] = 0$ for subspace $H \subset \mathcal{S}_k$ and $\mathrm{Dim}[H] < d_k$.*

[1] Continuous distribution here indicates that the data distribution is non-degenerate in the sense that the probability measure of any hyperplane of dimension less than that of the subspace is 0.

Claim 2. *Under the assumptions of Theorem 8, with probability* 1*, any* $L \leq d_k$ *points in the data* $\boldsymbol{X}^{(k)} \in \mathbb{R}^{d \times n_k}$ *that lie in* $\mathcal{S}_k$ *are linearly independent.*

Proof. For any set $\{\mathbf{x}_{j_\ell}\}_{\ell=1}^{L} \subseteq \boldsymbol{X}^{(k)}$ that are linearly dependent, let $H_{\mathbf{A}}$ be the subspace spanned by point set $\mathbf{A}$. Then at least one point in $\{\mathbf{x}_{j_\ell}\}_{\ell=1}^{L} \subseteq \boldsymbol{X}^{(k)}$ can be linearly represented by the others, and

$$\begin{aligned} &\Pr[\{\mathbf{x}_{j_\ell}\}_{\ell=1}^{L} \colon \{\mathbf{x}_{j_\ell}\}_{\ell=1}^{L} \text{ are linearly dependent}] \\ &\leq \sum_{\ell'=1}^{L} \Pr[\mathbf{x}_{j_{\ell'}} \in H_{\{\mathbf{x}_{j_\ell}^{-\ell'}\}}] = 0 \end{aligned} \tag{A.40}$$

where $\{\mathbf{x}_{j_\ell}^{-\ell'}\}$ indicates all the elements of $\{\mathbf{x}_{j_\ell}\}_{\ell=1}^{L}$ except $\mathbf{x}_{j_{\ell'}}$.

Since $\mathrm{Dim}[H_{\{\mathbf{x}_{j_\ell}^{(-\ell')}\}}] < L \leq d_k$, $\Pr[\mathbf{x}_{j_{\ell'}} \in H_{\{\mathbf{x}_{j_\ell}^{-\ell'}\}}] = 0$ for each $1 \leq \ell' \leq L$. □

Proof. According to Claim 2, for any fixed $1 \leq k \leq K$, any $L \leq d_k$ points in the data $\boldsymbol{X}^{(k)} \in \mathbb{R}^{d \times n_k}$ are almost surely linearly independent. Therefore, at least d_k points in $\boldsymbol{X}^{(k)}$ are required to linearly represent any point $\mathbf{x}_i$ in $\mathcal{S}_k$. Let $\boldsymbol{\alpha}^{i*}$ be the optimal solution to the following ℓ^0 problem

$$\min_{\boldsymbol{\alpha}^i} \|\boldsymbol{\alpha}^i\|_0 \quad s.t.\ \mathbf{x}_i = [\boldsymbol{X}^{(k)} \setminus \mathbf{x}_i \quad \boldsymbol{X}^{(-k)}]\boldsymbol{\alpha}^i,\ \boldsymbol{\alpha}_{ii} = 0 \tag{A.41}$$

where $\boldsymbol{X}^{(-k)}$ denotes the data that lie in all subspaces except $\mathcal{S}_k$. Let $\boldsymbol{\alpha}^{i*} = \begin{bmatrix} \boldsymbol{\beta}^* \\ \boldsymbol{\beta}^{-1*} \end{bmatrix}$ where $\boldsymbol{\beta}^*$ and $\boldsymbol{\beta}^{-1*}$ are sparse codes corresponding to $\boldsymbol{X}^{(k)} \setminus \mathbf{x}_i$ and $\boldsymbol{X}^{(-k)}$ respectively. Suppose $\boldsymbol{\beta}^{-1*} \neq \mathbf{0}$, then $\mathbf{x}_i$ belongs to a subspace $\mathcal{S}^{'}$ spanned by the data points corresponding to nonzero elements of $\boldsymbol{\alpha}^{i*}$, and $\mathcal{S}^{'} \neq \mathcal{S}_k$, $\mathrm{Dim}[\mathcal{S}^{'}] \leq d_k$. To see this, if $\mathcal{S}^{'} = \mathcal{S}_k$, then the data corresponding to nonzero elements of $\boldsymbol{\beta}^{-1*}$ belong to $\mathcal{S}_k$, which is contrary to the definition of $\boldsymbol{X}^{(-k)}$. Also, if $\mathrm{Dim}[\mathcal{S}^{'}] > d_k$, then a sparser solution can be obtained within $\boldsymbol{X}^{(k)}$, i.e. one can find d_k points in $\mathcal{S}_k$ to represent $\mathbf{x}_i$ almost surely.

Let $\mathcal{S}^{''} = \mathcal{S}^{'} \cap \mathcal{S}_k$, then $\mathrm{Dim}[\mathcal{S}^{''}] \leq d_k$. $\mathcal{S}^{''}$ is "inter-subspace hyperplane" since it intersects with at least two subspaces. We now derive the following results according to dimension of $\mathcal{S}^{''}$:

- $\mathrm{Dim}[\mathcal{S}^{''}] < d_k$. For each configuration of the generated data

$$\{\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n\},$$

$\mathcal{S}^{''}$ is the intersection of $\mathcal{S}_k$ and $\mathcal{S}^{'}$. A configuration of the data is a specific set of data points generated from the corresponding distributions. $\mathcal{S}^{'}$ can only be spanned from a subset of these data points, so there are only finite possible choices for $\mathcal{S}^{'}$ regardless of $\mathbf{x}_i$, and there are also finite possible choices for the hyperplane $\mathcal{S}^{''}$. According to Claim 1, the probability of the event that $\mathbf{x}_i$ lies in the hyperplane $\mathcal{S}^{''}$ is zero, i.e. $\Pr[\mathbf{x}_i \in \mathcal{S}^{''}|\{\mathbf{x}_j\}_{j\neq i}] = 0$. Now we compute the integral of this probability over the domain of $\{\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n\}$ (their corresponding subspaces) with respect to their corresponding probabilistic measures, we conclude that the probability that $\mathbf{x}_i \in \mathcal{S}^{''}$ is zero, i.e.

$$\begin{aligned}\Pr[\mathbf{x}_i \in \mathcal{S}^{''}] &= \int_{\times_{t=1}^{n}\mathcal{S}^{(t)}} \mathbb{1}_{\mathbf{x}_i\in\mathcal{S}^{''}} \otimes_{t=1}^{n} d\mu^{(t)} \\ &= \int_{\times_{t\neq i}\mathcal{S}^{(t)}} \Pr[\mathbf{x}_i \in \mathcal{S}^{''}|\{\mathbf{x}_t\}_{t\neq i}] \otimes_{t\neq i} d\mu^{(t)} = 0\end{aligned}$$

where $\mathcal{S}^{(t)}$ is the subspace that $\mathbf{x}_t$ lies in, and $\mu^{(t)}$ is the probabilistic measure of the distribution in $\mathcal{S}^{(t)}$.

- $\mathrm{Dim}[\mathcal{S}^{''}] = d_k$. In this case, $\mathcal{S}^{''} = \mathcal{S}^{'} = \mathcal{S}_k$, which indicates that the data points corresponding to nonzero elements of $\boldsymbol{\beta}^{-1^*}$ belong to $\mathcal{S}_k$, contradicting with the definition of $\boldsymbol{X}^{(-k)}$.

Therefore, with probability 1, $\boldsymbol{\beta}^{-1^*} = \mathbf{0}$, and the conclusion of Theorem 8 holds. □

A.2.2 Discussion of the Assumptions on the Subspaces

The only assumption on the subspaces in Theorem 8 is that all subspaces are distinct, which is the mildest assumption on the underlying subspaces compared to most existing sparse subspace clustering methods. Note that the difference between assumption S_3, i.e. overlapping subspaces, and assumption S_4 in Table 4.1, i.e. distinct subspaces, is that distinctness of subspaces allows the case that one small subspace $\mathcal{S}_k$ is contained in another big subspace $\mathcal{S}_{k'}$. ℓ^0-induced sparse subspace clustering can even produce subspace spare representation for the points in the small subspace, i.e. the nonzero elements of the optimal solution to the ℓ^0 problem (A.41) for any point $\mathbf{x}_i \in \mathcal{S}_k$ only correspond to data in subspace $\mathcal{S}_k$. One can intuitively obtain this result by noting that $\mathrm{Dim}[\mathcal{S}_k] = d_k < \mathrm{Dim}[\mathcal{S}_{k'}] = d_{k'}$,

otherwise $\mathcal{S}_k = \mathcal{S}_{k'}$, and it contradicts the assumption that $\mathcal{S}_k \neq \mathcal{S}_{k'}$. Also, d_k points in $\mathcal{S}_k$ other than $\mathbf{x}_i$ can linearly represent $\mathbf{x}_i$ almost surely, which forms the most sparse representation of $\mathbf{x}_i$ and constitutes the solution to the problem (A.41). In contrast, with probability 1, at least $d_{k'} > d_k$ points from $\boldsymbol{X}^{k'}$ other than $\mathbf{x}_i$ are needed to linearly represent $\mathbf{x}_i$ (note that the probability that a point from $\boldsymbol{X}^{k'}$ lies in a low dimensional subspace $\mathcal{S}_k$ is zero). Figure A.2 illustrates the example that a two-dimensional subspace $\mathcal{S}_1$ is contained in a three dimensional subspace $\mathcal{S}_2$. Two points $\mathbf{x}_2$ and $\mathbf{x}_3$ in $\mathcal{S}_1$ can linearly represent $\mathbf{x}_1 \in \mathcal{S}_1$, while at least three points in $\mathcal{S}_2$ are required to linear represent $\mathbf{x}_1$ with probability 1 almost surely. Although it is possible that two points in $\mathcal{S}_2$ can linear represent $\mathbf{x}_1$, the probability that this event happens is 0.

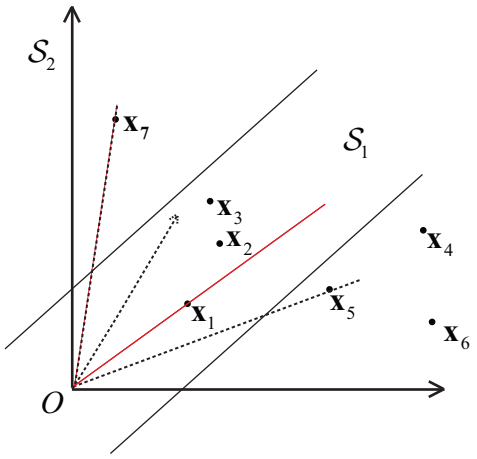


Figure A.2: A two-dimensional subspace $\mathcal{S}_1$ (a plane) is contained in a three dimensional subspace $\mathcal{S}_2$. $\mathbf{x}_1$ lies in $\mathcal{S}_1$, two points $\mathbf{x}_2$ and $\mathbf{x}_3$ in $\mathcal{S}_1$ can linearly represent $\mathbf{x}_1$. With probability 1, at least three points in $\mathcal{S}_2$, e.g. $\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6$, are required to linear represent $\mathbf{x}_1$. Note that it is possible that two points $\mathbf{x}_5$ and $\mathbf{x}_7 \in \mathcal{S}_2$ can linear represent $\mathbf{x}_1$, but it happens only if $\mathbf{x}_1$ lies in the red line which is the intersection of the plane $\mathcal{S}_1$ and the plane spanned by $\mathbf{x}_5$ and $\mathbf{x}_7$, and the probability of such event is 0.

A.2.3 Proof of Theorem 9

Theorem 9. *(Algorithm that renders subspace representation solves ℓ^0 sparse representation) Under the assumptions of Theorem 8, if there is an algorithm which, for any data point $\mathbf{x}_i \in \mathcal{S}_k$, $i \in [n], k \in [K]$, can find the data from*

the same subspace as $\mathbf{x}_i$ *that linearly represent* $\mathbf{x}_i$*, i.e.*

$$\mathbf{x}_i = \boldsymbol{X}\boldsymbol{\beta} \quad (\boldsymbol{\beta}_i = 0) \tag{A.42}$$

where nonzero elements of $\boldsymbol{\beta}$ *correspond to the data that lie in the subspace* $\mathcal{S}_k$*, then, with probability* 1*, the solution to the* ℓ^0 *problem (A.41) can be obtained from* $\boldsymbol{\beta}$ *in* $\mathcal{O}(\hat{n}^3)$ *time, where* $\hat{n}$ *is the number of nonzero elements in* $\boldsymbol{\beta}$*.*

Proof. Let $\hat{\boldsymbol{X}}$ be the data corresponding to the nonzero elements of $\boldsymbol{\beta}$. By Gaussian elimination, the maximal linearly independent columns of $\hat{\boldsymbol{X}}$, denoted by $\tilde{\boldsymbol{X}}$, can be obtained in $\mathcal{O}(\hat{n}^3)$ time where $\hat{n}$ is the number of columns of $\hat{\boldsymbol{X}}$. Then, $\mathbf{x}_i$ can be linearly represented by $\tilde{\boldsymbol{X}}$ and suppose $\mathbf{x}_i = \boldsymbol{X}\tilde{\boldsymbol{\beta}}$ where nonzero elements of $\tilde{\boldsymbol{\beta}}$ correspond to columns of $\tilde{\boldsymbol{X}}$. Then we will prove that $\tilde{\boldsymbol{\beta}}$ is the solution to the ℓ^0 problem (A.41) with probability 1. To see this, suppose $\tilde{\boldsymbol{\beta}}$ is not the sparest solution to (A.41), and denote by $\boldsymbol{\beta}^*$ the optimal solution to (A.41). Then $\mathbf{x}_i = \boldsymbol{X}\boldsymbol{\beta}^*$ and $\|\boldsymbol{\beta}^*\|_0 < \|\tilde{\boldsymbol{\beta}}\|_0$.

Since $\mathbf{x}_i$ lies in subspace $\mathcal{S}_k$, $d^* \triangleq \|\boldsymbol{\beta}^*\|_0 < \|\tilde{\boldsymbol{\beta}}\|_0 \leq d_k$ with probability 1. Let $\boldsymbol{X}^* = \{\mathbf{x}_{j_m}\}_{m=1}^{d^*}$ be the d^* data points corresponding to nonzero elements of $\boldsymbol{\beta}^*$. Then $\boldsymbol{X}^*$ must be linearly independent; otherwise, a sparser solution to (A.41) can be obtained by searching for the maximal linearly independent subset of $\boldsymbol{X}^*$. Denote by $\mathcal{S}^*$ the subspace spanned by $\boldsymbol{X}^*$ with $\mathrm{Dim}[\mathcal{S}^*] = d^*$, and $\mathcal{S}^{'} = \mathcal{S}^* \cap \mathcal{S}_k$. It follows that $\mathcal{S}^{'}$ is a subspace contained in $\mathcal{S}_k$ with dimensionality $\mathrm{Dim}[\mathcal{S}^{'}] \leq \mathrm{Dim}[\mathcal{S}^*] < d_k$. Note that the probability that $\mathbf{x}_i \in \mathcal{S}^{'}$ is zero since $\mathcal{S}^{'}$ is a low dimensional subspace in $\mathcal{S}_k$ and $\mathbf{x}_i$ is distributed according to continuous distribution supported on $\mathcal{S}_k$. □

A.2.4 Proof of Lemma 7

Before proving Lemma 7, we review the iterative proximal method for optimizing ℓ^0-graph, which obtains $\boldsymbol{\alpha}^{i(t)}$ from $\boldsymbol{\alpha}^{i(t-1)}$ for $t \geq 1$ by the following two steps:

$$\tilde{\boldsymbol{\alpha}^{i}}^{(t)} = \boldsymbol{\alpha}^{i(t-1)} - \frac{2}{\tau s}(\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\alpha}^{i(t-1)} - \boldsymbol{X}^\top \boldsymbol{X}) \tag{A.43}$$

$$\boldsymbol{\alpha}_j^{i^{(t)}} = \begin{cases} 0 & : \ |\tilde{\boldsymbol{\alpha}}_j^{i^{(t)}}| < \sqrt{\frac{2\lambda}{\tau s}} \text{ or } i = j \\ \tilde{\boldsymbol{\alpha}}_j^{i^{(t)}} & : \ \text{otherwise} \end{cases} \tag{A.44}$$

In the following text, we let $\sigma_{\max}(\cdot)$ and $\sigma_{\min}(\cdot)$ indicate the largest and smallest eigenvalues of a matrix in magnitude.

Lemma 7. *(Support Shrinkage in the Proximal Iterations and Sufficient Decrease of the Objective) When $s > \max\{2A_i, \frac{2(1+\lambda A_i)}{\lambda\tau}\}$, then the sequence $\{\boldsymbol{\alpha}^{i^{(t)}}\}_t$ generated by the proximal method with (A.43) and (A.44) satisfies*

$$\operatorname{supp}(\boldsymbol{\alpha}^{i^{(t)}}) \subseteq \operatorname{supp}(\boldsymbol{\alpha}^{i^{(t-1)}}), t \geq 1 \tag{A.45}$$

namely the support of the sequence $\{\boldsymbol{\alpha}^{i^{(t)}}\}_t$ shrinks when the iterative proximal proceeds. Moreover, the sequence of the objective $\{L(\boldsymbol{\alpha}^{i^{(t)}})\}_t$ decreases, and the following inequality holds for $t \geq 1$:

$$L(\boldsymbol{\alpha}^{i^{(t)}}) \leq L(\boldsymbol{\alpha}^{i^{(t-1)}}) - \frac{(\tau-1)s}{2}\|\boldsymbol{\alpha}^{i^{(t)}} - \boldsymbol{\alpha}^{i^{(t-1)}}\|_2^2 \tag{A.46}$$

And it follows that the sequence $\{L(\boldsymbol{\alpha}^{i^{(t)}})\}_t$ converges. The above results hold for any $1 \leq i \leq n$.

Proof. We prove this Lemma by mathematical induction.

When $t = 1$, we first show that $\operatorname{supp}(\boldsymbol{\alpha}^{i^{(1)}}) \subseteq \operatorname{supp}(\boldsymbol{\alpha}^{i^{(0)}})$, i.e. the support of $\boldsymbol{\alpha}^i$ shrinks after the first iteration. To see this, $\tilde{\boldsymbol{\alpha}}^{i^{(t)}} = \boldsymbol{\alpha}^{i^{(t-1)}} - \frac{2}{\tau s}(\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{\alpha}^{i^{(t-1)}} - \boldsymbol{X}^\top\boldsymbol{X})$. Since $\boldsymbol{\alpha}^{i^{(t-1)}} = \arg\min_{\boldsymbol{\alpha}^i \in \mathbb{R}^n, \boldsymbol{\alpha}_i^i = 0} \|\mathbf{x}_i - \boldsymbol{X}\boldsymbol{\alpha}^i\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$ is the optimal solution to the ℓ^1-graph problem, and the data are normalized to have unit ℓ^2-norm,

$$\|\mathbf{x}_i - \boldsymbol{X}\boldsymbol{\alpha}^{i^{(t-1)}}\|_2^2 + \lambda\|\boldsymbol{\alpha}^{i^{(t-1)}}\|_1 \leq \|\mathbf{x}_i\|_2^2 = 1$$

which indicates that $\|\mathbf{x}_i - \boldsymbol{X}\boldsymbol{\alpha}^{i^{(t-1)}}\|_2^2 \leq 1$. Letting $\mathbf{g}^{(t-1)} = -\frac{2}{\tau s}(\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{\alpha}^{i^{(t-1)}} - \boldsymbol{X}^\top\mathbf{x}_i)$, then

$$|\tilde{\boldsymbol{\alpha}}_j^{i^{(t)}}| \leq \|\mathbf{g}^{(t-1)}\|_\infty \leq \frac{2}{\tau s}\|\boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{\alpha}^{i^{(t-1)}} - \mathbf{x}_i)\|_\infty \leq \frac{2}{\tau s}$$

where j is the index for any zero element of $\boldsymbol{\alpha}^{i(t-1)}$, $1 \leq j \leq n, j \notin \text{supp}(\boldsymbol{\alpha}^{i(t-1)})$. Now $|\tilde{\boldsymbol{\alpha}_j^i}^{(t)}| < \sqrt{\frac{2\lambda}{\tau s}}$, and it follows that and $\boldsymbol{\alpha}_j^{i(t)} = 0$ due to the update rule in (A.44). Therefore, the zero elements of $\boldsymbol{\alpha}^{i(t-1)}$ remain unchanged in $\boldsymbol{\alpha}^{i(t)}$, and $\text{supp}(\boldsymbol{\alpha}^{i(t)}) \subseteq \text{supp}(\boldsymbol{\alpha}^{i(t-1)})$ for $t = 1$.

Letting $Q_{\mathbf{S}_i}(\mathbf{y}) = \|\mathbf{x}_i - \boldsymbol{X}_{\mathbf{S}_i}\mathbf{y}\|_2^2$ for $\mathbf{y} \in \mathbb{R}^{A_i}$, we show that $s > 2A_i$ is the Lipschitz constant for the gradient of function $Q_{\mathbf{S}_i}$. To see this, we have

$$\sigma_{\max}(\boldsymbol{X}_{\mathbf{S}_i}^\top \boldsymbol{X}_{\mathbf{S}_i}) = \big(\sigma_{\max}(\boldsymbol{X}_{\mathbf{S}_i})\big)^2 \leq \text{Tr}(\boldsymbol{X}_{\mathbf{S}_i}^\top \boldsymbol{X}_{\mathbf{S}_i}) = A_i$$

Also, $\nabla Q_{\mathbf{S}_i}(\mathbf{y}) = 2(\boldsymbol{X}_{\mathbf{S}_i}^\top \boldsymbol{X}_{\mathbf{S}_i}\mathbf{y} - \boldsymbol{X}_{\mathbf{S}_i}^\top \mathbf{x}_i)$, and

$$\begin{aligned} &\|\nabla Q_{\mathbf{S}_i}(\mathbf{y}) - \nabla Q_{\mathbf{S}_i}(\mathbf{z})\|_2 = 2\|\boldsymbol{X}_{\mathbf{S}_i}^\top \boldsymbol{X}_{\mathbf{S}_i}(\mathbf{y} - \mathbf{z})\|_2 \\ &\leq 2\sigma_{\max}(\boldsymbol{X}_{\mathbf{S}_i}^\top \boldsymbol{X}_{\mathbf{S}_i}) \cdot \|(\mathbf{y} - \mathbf{z})\|_2 \\ &\leq 2A_i\|(\mathbf{y} - \mathbf{z})\|_2 < s\|(\mathbf{y} - \mathbf{z})\|_2 \end{aligned} \tag{A.47}$$

Note that when $t = 1$, since

$$\boldsymbol{\alpha}^{i(t)} = \underset{\mathbf{v} \in \mathbb{R}^n, \mathbf{v}_i = 0}{\arg\min} \frac{\tau s}{2}\|\mathbf{v} - \tilde{\boldsymbol{\alpha}^i}^{(t)}\|_2^2 + \lambda\|\mathbf{v}\|_0$$

we have

$$\begin{aligned} &\frac{\tau s}{2}\|\boldsymbol{\alpha}^{i(t)} - \tilde{\boldsymbol{\alpha}^i}^{(t)}\|_2^2 + \lambda\|\boldsymbol{\alpha}^{i(t)}\|_0 \\ &\leq \frac{\tau s}{2}\|\frac{\nabla Q(\boldsymbol{\alpha}^{i(t-1)})}{\tau s}\|_2^2 + \lambda\|\boldsymbol{\alpha}^{i(t-1)}\|_0 \end{aligned} \tag{A.48}$$

which is equivalent to

$$\begin{aligned} &\langle \nabla Q_{\mathbf{S}_i}(\boldsymbol{\alpha}_{\mathbf{S}_i}^{i\,(t-1)}), \boldsymbol{\alpha}_{\mathbf{S}_i}^{i\,(t)} - \boldsymbol{\alpha}_{\mathbf{S}_i}^{i\,(t-1)} \rangle + \frac{\tau s}{2}\|\boldsymbol{\alpha}^{i(t)} - \boldsymbol{\alpha}^{i(t-1)}\|_2^2 \\ &+ \lambda\|\boldsymbol{\alpha}^{i(t)}\|_0 \leq \lambda\|\boldsymbol{\alpha}^{(t-1)}\|_0 \end{aligned} \tag{A.49}$$

due to the fact that

$$\langle \nabla Q(\boldsymbol{\alpha}^{i(t-1)}), \boldsymbol{\alpha}^{i(t)} - \boldsymbol{\alpha}^{i(t-1)} \rangle = \langle \nabla Q_{\mathbf{S}_i}(\boldsymbol{\alpha}_{\mathbf{S}_i}^{i\,(t-1)}), \boldsymbol{\alpha}_{\mathbf{S}_i}^{i\,(t)} - \boldsymbol{\alpha}_{\mathbf{S}_i}^{i\,(t-1)} \rangle$$

Also, since s is the Lipschitz constant for $\nabla Q_{\mathbf{S}_i}$,

$$Q_{\mathbf{S}_i}(\boldsymbol{\alpha}^{i}_{\mathbf{S}_i}{}^{(t)}) \leq Q_{\mathbf{S}_i}(\boldsymbol{\alpha}^{i}_{\mathbf{S}_i}{}^{(t-1)}) + \langle \nabla Q_{\mathbf{S}_i}(\boldsymbol{\alpha}^{i}_{\mathbf{S}_i}{}^{(t-1)}), \boldsymbol{\alpha}^{i}_{\mathbf{S}_i}{}^{(t)} - \boldsymbol{\alpha}^{i}_{\mathbf{S}_i}{}^{(t-1)} \rangle \tag{A.50}$$
$$+ \frac{s}{2} \|\boldsymbol{\alpha}^{i}_{\mathbf{S}_i}{}^{(t)} - \boldsymbol{\alpha}^{i}_{\mathbf{S}_i}{}^{(t-1)}\|_2^2$$

Combining (A.49) and (A.50) and noting that $\|\boldsymbol{\alpha}^{i}_{\mathbf{S}_i}{}^{(t)} - \boldsymbol{\alpha}^{i}_{\mathbf{S}_i}{}^{(t-1)}\|_2 = \|\boldsymbol{\alpha}^{i(t)} - \boldsymbol{\alpha}^{i(t-1)}\|_2$, $Q_{\mathbf{S}_i}(\boldsymbol{\alpha}^{i}_{\mathbf{S}_i}{}^{(t)}) = Q(\boldsymbol{\alpha}^{i(t)})$ and $Q_{\mathbf{S}_i}(\boldsymbol{\alpha}^{i}_{\mathbf{S}_i}{}^{(t-1)}) = Q(\boldsymbol{\alpha}^{i(t-1)})$, we have

$$Q(\boldsymbol{\alpha}^{i(t)}) + \lambda \|\boldsymbol{\alpha}^{i(t)}\|_0 \leq Q(\boldsymbol{\alpha}^{i(t-1)}) + \lambda \|\boldsymbol{\alpha}^{i(t-1)}\|_0 \tag{A.51}$$
$$- \frac{(\tau - 1)s}{2} \|\boldsymbol{\alpha}^{i(t)} - \boldsymbol{\alpha}^{i(t-1)}\|_2^2$$

Now (A.45) and (A.46) are verified for $t = 1$. Suppose (A.45) and (A.46) hold for all $t \geq t_0$ with $t_0 \geq 1$. Since $\{L(\boldsymbol{\alpha}^{i(t)})\}_{t=1}^{t_0}$ is decreasing, we have

$$L(\boldsymbol{\alpha}^{i(t_0)}) = \|\mathbf{x}_i - \boldsymbol{X}\boldsymbol{\alpha}^{i(t_0)}\|_2^2 + \lambda \|\boldsymbol{\alpha}^{i(t_0)}\|_0$$
$$\leq \|\mathbf{x}_i - \boldsymbol{X}\boldsymbol{\alpha}^{i(0)}\|_2^2 + \lambda \|\boldsymbol{\alpha}^{i(0)}\|_0 \leq 1 + \lambda A_i$$

which indicates that $\|\mathbf{x}_i - \boldsymbol{X}\boldsymbol{\alpha}^{i(t_0)}\|_2 \leq \sqrt{1 + \lambda A_i}$. When $t = t_0 + 1$,

$$|\tilde{\boldsymbol{\alpha}}^{i}_{j}{}^{(t)}| \leq \|\mathbf{g}^{(t-1)}\|_\infty \leq \frac{2}{\tau s} \|\boldsymbol{X}^\top (\boldsymbol{X}\boldsymbol{\alpha}^{i(t-1)} - \mathbf{x}_i)\|_\infty$$
$$\leq \frac{2}{\tau s} \sqrt{1 + \lambda A_i}$$

where j is the index for any zero element of $\boldsymbol{\alpha}^{i(t-1)}$, $1 \leq j \leq n, j \notin \mathrm{supp}(\boldsymbol{\alpha}^{i(t-1)})$. Now $|\tilde{\boldsymbol{\alpha}}^{i}_{j}{}^{(t)}| < \sqrt{\frac{2\lambda}{\tau s}}$, and it follows that and $\boldsymbol{\alpha}^{i}_{j}{}^{(t)} = 0$ due to the update rule in (A.44). Therefore, the zero elements of $\boldsymbol{\alpha}^{i(t-1)}$ remain unchanged in $\boldsymbol{\alpha}^{i}_{j}{}^{(t)}$, and $\mathrm{supp}(\boldsymbol{\alpha}^{i(t)}) \subseteq \mathrm{supp}(\boldsymbol{\alpha}^{i(t-1)}) \subseteq \mathbf{S}_i$ for $t = t_0 + 1$. Moreover, similar to the case when $t = 1$, we can derive (A.49), (A.50) and (A.51), so that the support shrinkage (A.45) and decline of the objective (A.46) are verified for $t = t_0 + 1$. It follows that the claim of this lemma holds for all $t \geq 1$.

Since the sequence $\{L(\boldsymbol{\alpha}^{i(t)})\}_t$ is decreasing with lower bound 0, it must converge. □

A.2.5 Proof of Lemma 8

In the following lemma, we show that the sequences $\{\boldsymbol{\alpha}^{i(t)}\}_t$ generated by the proximal method with (A.43) and (A.44) converges to a critical point of $L(\boldsymbol{\alpha}^i)$, which is denoted by $\hat{\boldsymbol{\alpha}}^i$. And we denote by $\boldsymbol{\alpha}^{i*}$ the global optimal solution to the ℓ^0-graph problem for point $\mathbf{x}_i$:

$$\min_{\boldsymbol{\alpha}^i \in \mathbb{R}^n, \boldsymbol{\alpha}^i_i = 0} L(\boldsymbol{\alpha}^i) = \|\mathbf{x}_i - \boldsymbol{X}\boldsymbol{\alpha}^i\|_2^2 + \lambda \|\boldsymbol{\alpha}^i\|_0 \tag{A.52}$$

Letting $\hat{\mathbf{S}}_i = \mathrm{supp}(\hat{\boldsymbol{\alpha}}^i)$, $\mathbf{S}_i^* = \mathrm{supp}(\hat{\boldsymbol{\alpha}}^i)$, the following lemma also shows that both $\hat{\boldsymbol{\alpha}}^i$ and $\boldsymbol{\alpha}^{i*}$ are local solutions to the capped-ℓ^1 regularized problem (4.15).

Lemma 8. *(Solution by our proximal method and the global optimal solution to the ℓ^0 problem are local solutions of capped-ℓ^1 regularized problem) For any $1 \leq i \leq n$, suppose $\kappa_-(A_i) > 0$; then the sequences $\{\boldsymbol{\alpha}^{i(t)}\}_t$ generated by the proximal method with (A.43) and (A.44) converges to a critical point of $L(\boldsymbol{\alpha}^i)$, which is denoted by $\hat{\boldsymbol{\alpha}}^i$. Moreover, if*

$$0 < b < \min\{\min_{j \in \hat{\mathbf{S}}_i} |\hat{\boldsymbol{\alpha}}^i_j|, \frac{\lambda}{\max_{j \notin \hat{\mathbf{S}}_i} |\frac{\partial Q}{\partial \boldsymbol{\alpha}^i_j}|_{\boldsymbol{\alpha}^i = \hat{\boldsymbol{\alpha}}^i}|},$$
$$\min_{j \in \mathbf{S}_i^*} |\boldsymbol{\alpha}^{i*}_j|, \frac{\lambda}{\max_{j \notin \mathbf{S}_i^*} |\frac{\partial Q}{\partial \boldsymbol{\alpha}^i_j}|_{\boldsymbol{\alpha}^i = \boldsymbol{\alpha}^{i*}}|}\}$$

(if the denominator is 0, $\frac{\lambda}{0}$ is defined to be $+\infty$ in this inequality), then both $\hat{\boldsymbol{\alpha}}^i$ and $\boldsymbol{\alpha}^{i*}$, i.e. the global optimal solution to (A.52), are local solutions to the capped-ℓ^1 regularized problem (4.15).

Proof. We first prove that the sequence $\{\boldsymbol{\alpha}^{i(t)}\}_t$ is bounded for any $1 \leq i \leq n$. In the proof of Lemma 7, it is proved that

$$\begin{aligned} L(\boldsymbol{\alpha}^{i(t)}) &= \|\mathbf{x}_i - \boldsymbol{X}\boldsymbol{\alpha}^{i(t)}\|_2^2 + \lambda \|\boldsymbol{\alpha}^{i(t)}\|_0 \\ &\leq \|\mathbf{x}_i - \boldsymbol{X}\boldsymbol{\alpha}^{i(0)}\|_2^2 + \lambda \|\boldsymbol{\alpha}^{i(0)}\|_0 \leq 1 + \lambda A_i \end{aligned}$$

for $t \geq 1$. Therefore, $\|\mathbf{x}_i - \boldsymbol{X}\boldsymbol{\alpha}^{i(t)}\|_2 \leq \sqrt{1 + \lambda A_i}$ and it follows that $\|\boldsymbol{X}\boldsymbol{\alpha}^{i(t)}\|_2^2 \leq (1 + \sqrt{1 + \lambda A_i})^2$. Since $\mathrm{supp}(\boldsymbol{\alpha}^{i(t)}) \subseteq \mathbf{S}_i$ for $t \geq 0$ due to Lemma 7,

$$\begin{aligned} &(1 + \sqrt{1 + \lambda A_i})^2 \geq \|\boldsymbol{X}\boldsymbol{\alpha}^{i(t)}\|_2 = \|\boldsymbol{X}_{\mathbf{S}_i} \boldsymbol{\alpha}^{i}_{\mathbf{S}_i}{}^{(t)}\|_2 \\ &\geq \sigma_{\min}(\boldsymbol{X}_{\mathbf{S}_i}{}^\top \boldsymbol{X}_{\mathbf{S}_i}) \|\boldsymbol{\alpha}^{i}_{\mathbf{S}_i}{}^{(t)}\|_2^2 = \sigma_{\min}(\boldsymbol{X}_{\mathbf{S}_i}{}^\top \boldsymbol{X}_{\mathbf{S}_i}) \|\boldsymbol{\alpha}^{i(t)}\|_2^2 \end{aligned}$$

Since $\kappa = \kappa_-(A_i) > 0$, we have $\sigma_{\min}({\boldsymbol{X}_{\mathbf{S}_i}}^\top \boldsymbol{X}_{\mathbf{S}_i}) \geq \kappa$ and it follows that ${\boldsymbol{\alpha}^i}^{(t)}$ is bounded: $\|{\boldsymbol{\alpha}^i}^{(t)}\|_2^2 \leq \frac{(1+\sqrt{1+\lambda A_i})^2}{\kappa}$. In addition, since ℓ^0-norm function $\|\cdot\|_0$ is a semi-algebraic function, therefore, according to Theorem 1 in [52], $\{{\boldsymbol{\alpha}^i}^{(t)}\}_t$ converges to a critical point of $L(\boldsymbol{\alpha}^i)$, denoted by $\hat{\boldsymbol{\alpha}^i}$.

Let $\hat{\mathbf{v}} = \boldsymbol{X}^\top(\boldsymbol{X}^\top\hat{\boldsymbol{\alpha}^i} - \mathbf{x}_i) + \lambda\dot{\mathbf{R}}(\hat{\boldsymbol{\alpha}^i}; b)$. For $j \in \hat{\mathbf{S}}_i$, since $\hat{\boldsymbol{\alpha}^i}$ is a critical point of $L(\boldsymbol{\alpha}^i) = \|\mathbf{x}_i - \boldsymbol{X}\boldsymbol{\alpha}^i\|_2^2 + \lambda\|\boldsymbol{\alpha}^i\|_0$, then $\frac{\partial Q}{\partial \boldsymbol{\alpha}^i_j}|_{\boldsymbol{\alpha}^i = \hat{\boldsymbol{\alpha}^i}} = 0$ because $\frac{\partial \|\boldsymbol{\alpha}^i\|_0}{\partial \boldsymbol{\alpha}^i_j}|_{\boldsymbol{\alpha}^i = \hat{\boldsymbol{\alpha}^i}} = 0$. Note that $\min_{j \in \hat{\mathbf{S}}_i} |\hat{\boldsymbol{\alpha}^i_j}| > b$, so $\frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}^i_j}|_{\boldsymbol{\alpha}^i = \hat{\boldsymbol{\alpha}^i}} = 0$, and it follows that $\hat{\mathbf{v}}_j = 0$.

For $j \notin \hat{\mathbf{S}}_i$, since $\frac{dR}{d\boldsymbol{\alpha}^i_j}(\hat{\boldsymbol{\alpha}}^i_j +; b) = \frac{\lambda}{b}$ and $\frac{dR}{d\boldsymbol{\alpha}^i_j}(\hat{\boldsymbol{\alpha}}^i_j -; b) = -\frac{\lambda}{b}$,

$\frac{\lambda}{b} > \max_{j \notin \hat{\mathbf{S}}_i} |\frac{\partial Q}{\partial \boldsymbol{\alpha}^i_j}|_{\boldsymbol{\alpha}^i = \hat{\boldsymbol{\alpha}^i}}|$, we can choose the j-th element of $\dot{\mathbf{R}}(\hat{\boldsymbol{\alpha}^i}; b)$ such that $\hat{\mathbf{v}}_j = 0$. Therefore, $\|\hat{\mathbf{v}}\|_2 = 0$, and $\hat{\boldsymbol{\alpha}^i}$ is a local solution to the problem (4.15).

Now we prove that ${\boldsymbol{\alpha}^i}^*$ is also a local solution to (4.15). Let $\mathbf{v}^* = \boldsymbol{X}^\top(\boldsymbol{X}^\top{\boldsymbol{\alpha}^i}^* - \mathbf{x}_i) + \lambda\dot{\mathbf{R}}({\boldsymbol{\alpha}^i}^*; b)$, and Q is defined as before. For $j \in \mathbf{S}_i^*$, since ${\boldsymbol{\alpha}^i}^*$ is the global optimal solution to problem (A.52), we also have $\frac{\partial Q}{\partial \boldsymbol{\alpha}^i_j}|_{\boldsymbol{\alpha}^i = {\boldsymbol{\alpha}^i}^*} = 0$. If it is not the case and $\frac{\partial Q}{\partial \boldsymbol{\alpha}^i_j}|_{\boldsymbol{\alpha}^i = {\boldsymbol{\alpha}^i}^*} \neq 0$, then we can change $\boldsymbol{\alpha}^i_j$ by a small amount in the direction of the gradient $\frac{\partial Q}{\partial \boldsymbol{\alpha}^i_j}$ at the point $\boldsymbol{\alpha}^i = {\boldsymbol{\alpha}^i}^*$ and still make $\boldsymbol{\alpha}^i_j \neq 0$, leading to a smaller value of the objective $L(\boldsymbol{\alpha}^i)$.

Note that $\min_{j \in \mathbf{S}_i^*} |{\boldsymbol{\alpha}^i_j}^*| > b$, so $\frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}^i_j}|_{\boldsymbol{\alpha}^i = \hat{\boldsymbol{\alpha}^i}} = 0$, and it follows that $\mathbf{v}^*_j = 0$.

For $j \notin \mathbf{S}_i^*$, since $\frac{\lambda}{b} > \max_{j \notin \hat{\mathbf{S}}_i} |\frac{\partial Q}{\partial \boldsymbol{\alpha}^i_j}|_{\boldsymbol{\alpha}^i = {\boldsymbol{\alpha}^i}^*}|$, we can choose the j-th element of $\dot{\mathbf{R}}({\boldsymbol{\alpha}^i}^*; b)$ such that $\mathbf{v}^*_j = 0$. It follows that $\|\mathbf{v}^*\|_2 = 0$, and ${\boldsymbol{\alpha}^i}^*$ is also a local solution to the problem (4.15). □

A.2.6 Proof of Theorem 10

Theorem 5 in [70] gives the estimation on the distances between two local solutions of the capped-ℓ^1 regularized problems, based on which we have the following theorem showing that the sub-optimal solution $\hat{\boldsymbol{\alpha}^i}$ obtained by our proximal method is close to the global optimal solution to the original ℓ^0 problem (A.52), i.e. ${\boldsymbol{\alpha}^i}^*$.

Theorem 10. *(Sub-optimal solution is close to the global optimal solution) For any $1 \leq i \leq n$, suppose $\kappa_-(A_i) > 0$ and $\kappa_-(|\hat{\mathbf{S}}_i \cup \mathbf{S}_i^*|) > \kappa > 0$, and b is chosen*

according to (5.24) as in Lemma 8. Then

$$\|\boldsymbol{X}(\hat{\boldsymbol{\alpha}}^i - \boldsymbol{\alpha}^{i*})\|_2^2 \leq \frac{2\kappa_-(|\hat{\mathbf{S}}_i \cup \mathbf{S}_i^*|)}{(\kappa_-(|\hat{\mathbf{S}}_i \cup \mathbf{S}_i^*|) - \kappa)^2} \tag{A.53}$$
$$\big(\sum_{j\in\hat{\mathbf{S}}_i}(\max\{0, \frac{\lambda}{b} - \kappa|\hat{\boldsymbol{\alpha}}_j^i - b|\})^2 + |\mathbf{S}_i^* \setminus \hat{\mathbf{S}}_i|(\max\{0, \frac{\lambda}{b} - \kappa b\})^2\big)$$

In addition,

$$\|(\hat{\boldsymbol{\alpha}}^i - \boldsymbol{\alpha}^{i*})\|_2^2 \leq \frac{2}{(\kappa_-(|\hat{\mathbf{S}}_i \cup \mathbf{S}_i^*|) - \kappa)^2} \tag{A.54}$$
$$\big(\sum_{j\in\hat{\mathbf{S}}_i}(\max\{0, \frac{\lambda}{b} - \kappa|\hat{\boldsymbol{\alpha}}_j^i - b|\})^2 + |\mathbf{S}_i^* \setminus \hat{\mathbf{S}}_i|(\max\{0, \frac{\lambda}{b} - \kappa b\})^2\big)$$

Proof. According to Lemma 8, both $\hat{\boldsymbol{\alpha}^i}$ and $\boldsymbol{\alpha}^{i*}$ are local solutions to problem (4.15). By Theorem 5 in [70], we have

$$\|\boldsymbol{X}(\hat{\boldsymbol{\alpha}^i} - \boldsymbol{\alpha}^{i*})\|_2^2 \leq \frac{2\kappa_-(|\hat{\mathbf{S}}_i \cup \mathbf{S}_i^*|)}{(\kappa_-(|\hat{\mathbf{S}}_i \cup \mathbf{S}_i^*|) - \kappa)^2}\big(\|\theta(|\hat{\boldsymbol{\alpha}}^i_{\hat{\mathbf{S}}_i}, \kappa)\|_2^2 \tag{A.55}$$
$$+ |\mathbf{S}_i^* \setminus \hat{\mathbf{S}}_i|\theta^2(0+, \kappa)\big)$$

By the definition of θ,

$$\theta(t, \kappa) = \sup_s\{-\mathrm{sgn}(s-t)(\dot{R}(s; b) - \dot{R}(t; b)) - \kappa|s - t|\}$$

Since $t > b$, it can be verified that $\theta(t, \kappa) = \max\{0, \frac{\lambda}{b} - \kappa|\hat{\boldsymbol{\alpha}}_j^i - b|\}$. Therefore,

$$\|\theta(|\hat{\boldsymbol{\alpha}}^i_{\hat{\mathbf{S}}_i}, \kappa)\|_2^2 = \sum_{j\in\hat{\mathbf{S}}_i}\big(\theta(\hat{\boldsymbol{\alpha}}_j^i, \kappa)\big)^2 \tag{A.56}$$
$$= \sum_{j\in\hat{\mathbf{S}}_i}(\max\{0, \frac{\lambda}{b} - \kappa|\hat{\boldsymbol{\alpha}}_j^i - b|\})^2 \tag{A.57}$$

It can also be verified that

$$\theta(0+, \kappa) = \max\{0, \frac{\lambda}{b} - \kappa b\} \tag{A.58}$$

So that (A.53) is proved. Let $\mathbf{S}' = \hat{\mathbf{S}}_i \cup \mathbf{S}_i^*$, since $\sigma_{\min}(\boldsymbol{X}_{\mathbf{S}'}^\top \boldsymbol{X}_{\mathbf{S}'}) \geq \kappa_-(|\hat{\mathbf{S}}_i \cup \mathbf{S}_i^*|)$, so that $\|\boldsymbol{X}(\hat{\boldsymbol{\alpha}^i} - \boldsymbol{\alpha}^{i*})\|_2^2 \geq \kappa_-(|\hat{\mathbf{S}}_i \cup \mathbf{S}_i^*|)\|(\hat{\boldsymbol{\alpha}^i} - \boldsymbol{\alpha}^{i*})\|_2^2$. It follows that (A.54) holds. □

Table A.1: Clustering Results on UMIST Face Data

UMIST Face / # Clusters	Measure	KM	SC	ℓ^1-Graph	SMCE	OMP-Graph	ℓ^0-Graph
c = 4	AC	0.4846	0.5691	0.4390	0.5203	0.4878	**0.5854**
	NMI	0.2919	0.4351	0.3303	0.3314	**0.4678**	0.4128
c = 8	AC	0.4347	0.4601	0.4930	0.4695	0.5211	**0.7042**
	NMI	0.5473	0.5087	0.5516	0.5744	0.5626	**0.7214**
c = 12	AC	0.4529	0.4805	0.5135	0.4955	0.5856	**0.6727**
	NMI	0.6216	0.6145	0.5972	0.6429	0.6615	**0.7615**
c = 16	AC	0.4278	0.4516	0.4562	0.4747	0.4885	**0.6175**
	NMI	0.6280	0.6455	0.6581	0.6909	0.5936	**0.7529**
c = 20	AC	0.4275	0.4052	0.4904	0.4487	0.4835	**0.6730**
	NMI	0.6426	0.6159	0.6885	0.6696	0.6310	**0.7924**

Table A.2: Clustering Results on CMU PIE Data

CMU PIE / # Clusters	Measure	KM	SC	ℓ^1-Graph	SMCE	OMP-Graph	ℓ^0-Graph
c = 20	AC	0.1320	0.1312	0.2291	0.2315	0.1076	**0.3306**
	NMI	0.1210	0.1302	0.2829	0.3071	0.0734	**0.4036**
c = 40	AC	0.1044	0.0880	0.2251	0.1903	0.0783	**0.3440**
	NMI	0.1522	0.1449	0.3257	0.3052	0.0914	**0.4626**
c = 68	AC	0.0845	0.0729	0.2287	0.1733	0.0821	**0.2591**
	NMI	0.1884	0.1789	0.3659	0.3343	0.1494	**0.4435**

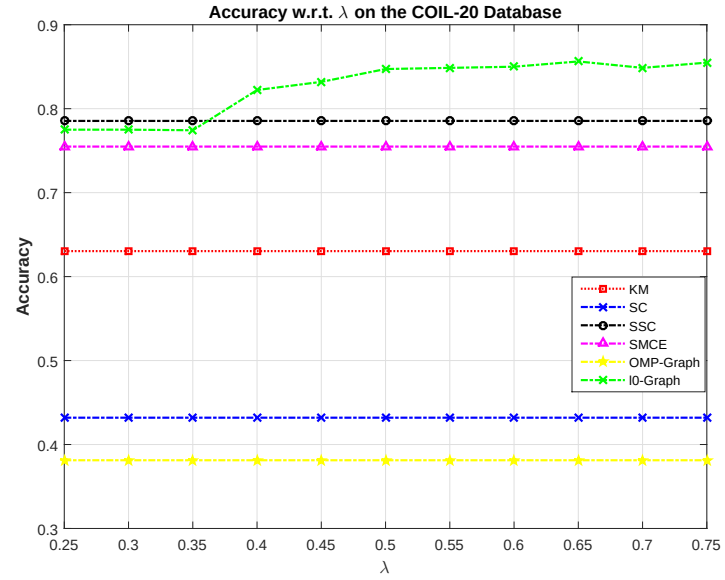
A.2.7 More Experimental Results

Parameter Sensitivity Result on the COIL-20 Database

We investigate how the clustering performance on the COIL-20 Database changes by varying the weighting parameter λ for ℓ^0-graph, and we illustrate the result in Figure A.3.

Additional Experimental Results

Table 4.6 shows the overall clustering performance of ℓ^0-graph on the UMIST Face Database and CMU PIE Face Database. We now show the detailed clustering performance on the first c clusters of this data set in Tables A.1 and A.2. The UMIST Face Database consists of 575 images of size 112×92 for 20 people. Each person is shown in a range of poses from profile to frontal views. CMU PIE face data contains cropped face images of size 32×32 for 68 persons, and there are around 170 facial images for each person under different illumination and expressions, with a total number of 11554 images.



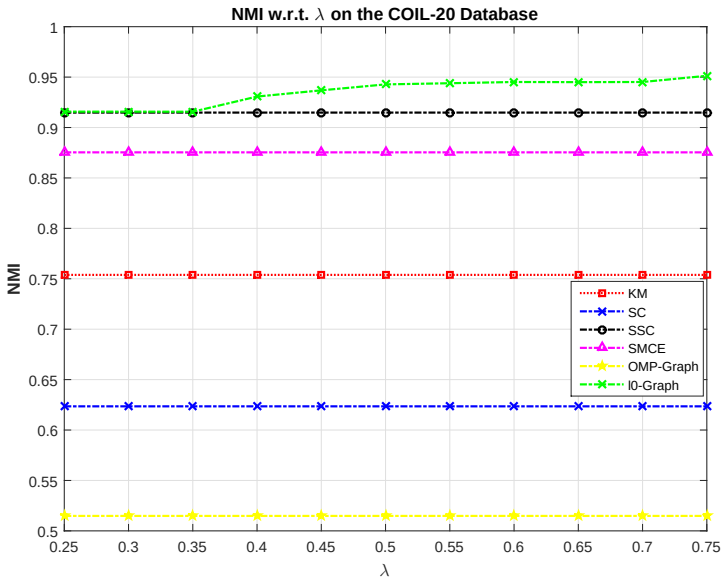


Figure A.3: Clustering performance with different values of λ, i.e. the weight for the ℓ^0-norm, on the COIL-20 Database. Left: Accuracy; Right: NMI.

REFERENCES

[1] H. Narayanan, M. Belkin, and P. Niyogi, "On the relation between low density separation, spectral clustering and graph cuts," in *NIPS*, 2006, pp. 1025–1032.

[2] S. Yan and H. Wang, "Semi-supervised learning by sparse representation," in *SDM*, 2009, pp. 792–801.

[3] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. S. Huang, "Learning with l1-graph for image analysis," *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 858–866, 2010.

[4] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, 2013. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2013.57

[5] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

[6] J. Liu, D. Cai, and X. He, "Gaussian mixture model with local consistency," in *AAAI*, 2010.

[7] X. He, D. Cai, Y. Shao, H. Bao, and J. Han, "Laplacian regularized Gaussian mixture model for data clustering," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 23, no. 9, pp. 1406–1418, Sept 2011.

[8] M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, and D. Cai, "Graph regularized sparse coding for image representation," *IEEE Transactions on Image Processing*, vol. 20, no. 5, pp. 1327–1336, 2011.

[9] S. Gao, I. W.-H. Tsang, and L.-T. Chia, "Laplacian sparse coding, hypergraph Laplacian sparse coding, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 92–104, 2013.

[10] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *NIPS*, 2001, pp. 849–856.

[11] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, 2003, pp. 912–919.

[12] J. A. Hartigan and M. A. Wong, "A K-means clustering algorithm," *Applied Statistics*, vol. 28, pp. 100–108, 1979.

[13] N. Shental, A. Zomet, T. Hertz, and Y. Weiss, "Pairwise clustering and graphical models," in *NIPS*, 2003.

[14] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, "Maximum margin clustering," in *NIPS*, 2004.

[15] Z. Karnin, E. Liberty, S. Lovett, R. Schwartz, and O. Weinstein, "Unsupervised SVMs: On the complexity of the furthest hyperplane problem," *Journal of Machine Learning Research - Proceedings Track*, vol. 23, pp. 2.1–2.17, 2012.

[16] R. Gomes, A. Krause, and P. Perona, "Discriminative clustering by regularized information maximization," in *NIPS*, 2010, pp. 775–783.

[17] M. Sugiyama, M. Yamada, M. Kimura, and H. Hachiya, "On information-maximization clustering: Tuning parameter selection and analytic solution," in *ICML*, 2011, pp. 65–72.

[18] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[19] L. Devroye, *A Probabilistic Theory of Pattern Recognition*. Springer, 1996, vol. 31.

[20] Y. Yang, "Minimax nonparametric classification - part i: Rates of convergence," *IEEE Transactions on Information Theory*, vol. 45, no. 7, pp. 2271–2284, 1999.

[21] J.-Y. Audibert and A. B. Tsybakov, "Fast learning rates for plug-in classifiers," *The Annals of Statistics*, vol. 35, no. 2, pp. pp. 608–633, 2007.

[22] R. Jenssen, D. Erdogmus, J. C. Príncipe, and T. Eltoft, "The Laplacian pdf distance: A cost function for clustering in a kernel feature space," in *NIPS*, 2004.

[23] E. Giné and A. Guillou, "Rates of strong uniform consistency for multivariate kernel density estimators," *Ann. Inst. H. Poincaré Probab. Statist.*, vol. 38, no. 6, pp. 907–921, Nov. 2002.

[24] U. Einmahl and D. M. Mason, "Uniform in bandwidth consistency of kernel-type function estimators," *The Annals of Statistics*, vol. 33, p. 1380C1403, 2005.

[25] R. M. Dudley, *Uniform Central Limit Theorems.* Cambridge University Press, 1999.

[26] A. van der Vaart and J. Wellner, *Weak Convergence and Empirical Processes*, ser. Springer series in statistics. Springer, 1996.

[27] D. Nolan and D. Pollard, "U-Processes: Rates of convergence," *The Annals of Statistics*, vol. 15, no. 2, 1987.

[28] O. Chapelle and A. Zien, "Semi-Supervised Classification by Low Density Separation," in *AISTATS*, 2005.

[29] M. Maier, U. von Luxburg, and M. Hein, "Influence of graph construction on graph-based clustering measures," in *NIPS*, 2008, pp. 1025–1032.

[30] Z. Xu, R. Jin, J. Zhu, I. King, M. R. Lyu, and Z. Yang, "Adaptive regularization for transductive support vector machine," in *NIPS*, 2009, pp. 2125–2133.

[31] X. Zhu, J. Lafferty, and R. Rosenfeld, "Semi-supervised learning with graphs," Ph.D. dissertation, Carnegie Mellon University, Language Technologies Institute, School of Computer Science, 2005.

[32] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972–977, 2007.

[33] D. Lashkari and P. Golland, "Convex clustering with exemplar-based models," in *NIPS*, 2007.

[34] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis, and density estimation," *Journal of the American Statistical Association*, vol. 97, no. 458, pp. 611–631, June 2002.

[35] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000.

[36] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, p. 2007, 2007.

[37] Y. Yang, X. Chu, F. Liang, and T. S. Huang, "Pairwise exemplar clustering," in *AAAI*, 2012.

[38] S. E. Schaeffer, "Survey: Graph clustering," *Comput. Sci. Rev.*, vol. 1, no. 1, pp. 27–64, Aug. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.cosrev.2007.05.001

[39] F. R. K. Chung, *Spectral Graph Theory*. American Mathematical Society, 1997.

[40] L. Mancera and J. Portilla, "L0-norm-based sparse representation through alternate projections," in *Image Processing, 2006 IEEE International Conference on*, Oct 2006, pp. 2089–2092.

[41] C. Bao, H. Ji, Y. Quan, and Z. Shen, "L0 norm based dictionary learning by proximal methods with global convergence," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2014. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2014.493 pp. 3858–3865.

[42] J. A. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004. [Online]. Available: http://dx.doi.org/10.1109/TIT.2004.834793

[43] J. Yang, K. Yu, Y. Gong, and T. S. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *CVPR*, 2009, pp. 1794–1801.

[44] H. Cheng, Z. Liu, L. Yang, and X. Chen, "Sparse representation and learning in visual recognition: Theory and applications," *Signal Process.*, vol. 93, no. 6, pp. 1408–1425, June 2013. [Online]. Available: http://dx.doi.org/10.1016/j.sigpro.2012.09.011

[45] T. Zhang, B. Ghanem, S. Liu, C. Xu, and N. Ahuja, "Low-rank sparse coding for image classification," in *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, 2013. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2013.42 pp. 281–288.

[46] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *CVPR*, 2009, pp. 2790–2797.

[47] Y. Wang and H. Xu, "Noisy sparse subspace clustering," in *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, 2013, pp. 89–97.

[48] M. Soltanolkotabi, E. Elhamifar, and E. J. Cands, "Robust subspace clustering," *The Annals of Statistics*, vol. 42, 04 2014.

[49] E. Elhamifar and R. Vidal, "Sparse manifold clustering and embedding," in *NIPS*, 2011, pp. 55–63.

[50] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[51] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[52] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Math. Program.*, vol. 146, no. 1-2, pp. 459–494, Aug. 2014. [Online]. Available: http://dx.doi.org/10.1007/s10107-013-0701-9

[53] X. Zheng, D. Cai, X. He, W.-Y. Ma, and X. Lin, "Locality preserving clustering for image database," in *Proceedings of the 12th Annual ACM International Conference on Multimedia*, ser. MULTIMEDIA '04. New York, NY, USA: ACM, 2004, pp. 885–891.

[54] D. Plummer and L. Lovász, *Matching Theory*, ser. North-Holland Mathematics Studies. Elsevier Science, 1986. [Online]. Available: http://books.google.com/books?id=mycZP-J344wC

[55] D. N. A. Asuncion, "UCI machine learning repository," in *University of California, Irvine, School of Information and Computer Sciences*, 2007.

[56] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image Vision Comput.*, vol. 28, no. 5, pp. 807–813, May 2010. [Online]. Available: http://dx.doi.org/10.1016/j.imavis.2009.08.002

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

[58] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, 2010, pp. 663–670.

[59] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[60] D. Park, C. Caramanis, and S. Sanghavi, "Greedy subspace clustering," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2014, pp. 2753–2761.

[61] Y.-X. Wang, H. Xu, and C. Leng, "Provable subspace clustering: When LRR meets SSC," in *Advances in Neural Information Processing Systems 26*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 64–72.

[62] M. Soltanolkotabi and E. J. Cands, "A geometric analysis of subspace clustering with outliers," *Ann. Statist.*, vol. 40, no. 4, pp. 2195–2238, 08 2012.

[63] Y. Yang, Z. Wang, J. Yang, J. Han, and T. Huang, "Regularized l1-graph for data clustering," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[64] X. Peng, Z. Yi, and H. Tang, "Robust subspace clustering via thresholding ridge regression," in *AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, 2015, pp. 3827–3833.

[65] R. Jenatton, J. Mairal, F. R. Bach, and G. R. Obozinski, "Proximal methods for sparse hierarchical dictionary learning," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 487–494.

[66] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Mar. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1756006.1756008

[67] J. Mairal, F. R. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning," in *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems*, 2008. [Online]. Available: http://papers.nips.cc/paper/3448-supervised-dictionary-learning pp. 1033–1040.

[68] E. L. Dyer, A. C. Sankaranarayanan, and R. G. Baraniuk, "Greedy feature selection for subspace clustering," *Journal of Machine Learning Research*, vol. 14, pp. 2487–2517, 2013.

[69] M. Hyder and K. Mahata, "An approximate l0 norm minimization algorithm for compressed sensing," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, April 2009, pp. 3365–3368.

[70] C.-H. Zhang and T. Zhang, "A general theory of concave regularization for high-dimensional sparse estimation problems," *Statist. Sci.*, vol. 27, no. 4, pp. 576–593, 11 2012.

[71] E. Candes and T. Tao, "Decoding by linear programming," *Information Theory, IEEE Transactions on*, vol. 51, no. 12, pp. 4203–4215, 2005.

[72] E. J. Cands, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 910, pp. 589 – 592, 2008.

[73] M. Karasuyama and H. Mamitsuka, "Manifold-based similarity adaptation for label propagation," in *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems*, 2013, pp. 1547–1555.

[74] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010. [Online]. Available: http://www.icml2010.org/papers/449.pdf pp. 399–406.

[75] G. Irie, Z. Li, X. Wu, and S. Chang, "Locally linear hashing for extracting non-linear manifolds," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, 2014. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2014.272 pp. 2123–2130.

[76] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *NIPS*, 2006, pp. 801–808.

[77] J. Mairal, F. R. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, 2009. [Online]. Available: http://doi.acm.org/10.1145/1553374.1553463 pp. 689–696.

[78] Z. Wang, J. Feng, and S. Yan, "Collaborative linear coding for robust image classification," *Int. J. Comput. Vis.*, vol. 114, p. 322333, 2015.