Learning Robust Graph Regularisation for Subspace Clustering

Elyor Kodirov
e.kodirov@qmul.ac.uk
Tao Xiang
t.xiang@qmul.ac.uk
Zhenyong Fu
z.fu@qmul.ac.uk
Shaogang Gong
s.gong@qmul.ac.uk

School of Electronic Engineering and Computer Science,
Queen Mary University of London,
London E1 4NS, UK

Abstract

Various subspace clustering methods have benefited from introducing a graph regularisation term in their objective functions. In this work, we identify two critical limitations of the graph regularisation term employed in existing subspace clustering models and provide solutions for both of them. First, the squared l_2-norm used in the existing term is replaced by a l_1-norm term to make the regularisation term more robust against outlying data samples and noise. Solving l_1 optimisation problems is notoriously expensive and a new formulation and an efficient algorithm are provided to make our model tractable. Second, instead of assuming that the graph topology and weights are known a priori and fixed during learning, we propose to learn the graph and integrate the graph learning into the proposed l_1-norm graph regularised optimisation problem. Extensive experiments conducted on five benchmark datasets show that the proposed robust subspace clustering method significantly outperforms the state-of-the-art.

1 Introduction

Given a set of unlabelled data points represented in a high-dimensional feature space, a subspace clustering model [9] aims to learn a lower-dimensional subspace where the intrinsic data distribution structure and data associations can be better revealed for easier clustering, compared to the original feature space. Although many different subspace clustering methods exist ([3, 7, 9, 16, 18, 19, 23]), they all consist of two steps: the first step involves subspace learning via some form of matrix factorisation and the second step groups the data into clusters in the learned subspace. The most common subspace learning model used is the self-expressive model which aims to explain the dataset using itself, e.g., the sparse representation for subspace clustering (SSC) model [9]. Beyond self-expressive models, various matrix factorisation models have been formulated for subspace learning. These include principal component analysis (PCA) [12], non-negative matrix factorisation (NMF) [13], and dictionary learning (DL) for sparse coding [22].

Recently it has been shown that regardless what subspace learning model is taken, a subspace clustering method can benefit from introducing a graph regularisation term in its objective function [10, 25]. This term is to enforce that the local data geometric structure is preserved in the learned subspace. For instance, [10] proposed to include graph regularisation in a self-expressive model. Similarly, [25] considered a combination of a low rank nuclear-norm on the self-expressive model and a graph regularisation term. The idea of graph regularisation is from the graph theory [6]. Formally, let $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ be a sparsely connected undirected graph between a set of data points where $\mathbf{V}$ is a set of graph vertices representing the data points and $\mathbf{E}$ the edge set. This graph can be encoded by an affinity matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ for N data points where $\mathbf{W}_{i,j} \neq 0$ if the two vertices are connected, *i.e.* the corresponding data points are in a local neighbourhood. The graph regularisation term $\Omega(\mathbf{Y})$ is defined as:

$$\Omega(\mathbf{Y}) = \sum_{ij}^{N} \mathbf{W}_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|_2^2, \tag{1}$$

where $\mathbf{y}$ is the data vector representation in the learned subspace.

Despite the popularity of exploiting graph regularisation for learning a better subspace for data clustering, there are two problems in the current formulation. (1) *The use of squared l_2-norm* – the first problem arises from the use of the squared l_2-norm in Eq. (1) to measure the distance between two data points $\mathbf{y}_i$ and $\mathbf{y}_j$ in the learned subspace. This is because a least squares based regularisation function can be easily dominated by outlying data samples. (2) *The construction of graph* – the second problem concerns with the space in which the graph should be constructed and how it is constructed. Most existing clustering models construct the graph $\mathbf{G}$ and compute the matrix $\mathbf{W}$ in the original high dimensional feature space where the raw data resides. Given that the data points in this high dimensional space are widely spread and unreliably represented (hence the learning of the subspace), the constructed graph is also suboptimal which does not reflect necessarily the true (and inherent) geometrical structure of the data distribution. This thus has an adverse effect on the subspace clustering. Most existing methods also assume that the graph is known a priori and fixed before subspace learning, whilst the optimal graph structure of the data is latent and needs to be learned jointly with the subspace.

In this paper, we present solutions to both problems. Specifically, for the first problem, to make the regularisation term more robust against outlying samples, we propose a graph weighted l_1-norm regularisation to suppress outlying samples. Note that when the squared l_2-norm is changed to l_1-norm, the optimisation problem becomes much harder to solve. We therefore propose a novel formulation for the graph regularisation term which can be solved efficiently. Furthermore, we propose to learn the optimal graph jointly with the proposed robust graph regularisation (RGR) in a single learning process. We consider that learning the optimal lower-dimensional representation $\mathbf{Y}$ and learning the optimal graph $\mathbf{G}$ on which $\mathbf{Y}$ is the most smooth are intrinsically related and cannot be modelled independently. Our overall framework for joint subspace learning, graph regularisation (local smoothness constraint in the lower-dimensional subspace), and clustering are formulated as a problem of dictionary learning for sparse coding followed by spectral clustering. Due to the two changes made to the standard graph regularisation term (Eq. (1)), our dictionary learning objective function is both non-smooth and non-convex. Solving this optimisation problem is thus non-trivial. To this end, we formulate an efficient iterative optimisation algorithm.

There are a few existing attempts to address the robustness and graph learning problems in isolation, but none of them addresses them jointly as we do. For the the graph learning

problem, [10] proposed a method to learn better graph weights in which data representation and graph weights are learned simultaneously. Using a different model, [20] proposed a method that solves graph weight matrix and clustering structure learning simultaneously by adaptively assigning neighbours for each data point based on local connectivity. Nevertheless, both methods try to solve the second problem only, while ignoring the first problem. In this paper, we argue that since these two problems are connected (they are represented in a single formulation, Eq. (1)), they should be solved jointly.

The contributions of this work are three-folds: (1) For the first time, a systematic analysis of the weakness of the widely used graph regularisation constraint in subspace clustering together with a solution is presented. (2) A novel dictionary learning for sparse coding model is formulated with a new robust l_1-norm graph regularisation term and graph learning. (3) An efficient iterative optimisation algorithm is presented which estimates jointly the unknown graph $\mathbf{G}$, the dictionary, and the low-dimensional representational space $\mathbf{Y}$. Extensive experiments are carried out on five widely used benchmark datasets for subspace clustering showing that our method outperforms significantly the state-of-the-art alternatives.

2 Methodology

2.1 Formulation

Given a data matrix $\mathbf{X} \in \mathbb{R}^{r \times N}$ with N r-dimensional data feature vectors as columns, dictionary learning for sparse coding aims to decompose $\mathbf{X}$ using d atoms of a learned dictionary $\mathbf{D} \in \mathbb{R}^{r \times d}$. This is as opposed to self-expressive sparse coding models [9] that use a predefined dictionary $\mathbf{D} = \mathbf{X}$. In its matrix form, the objective function can be written as:

$$\min_{\mathbf{D},\mathbf{Y}} \|\mathbf{X} - \mathbf{D}\mathbf{Y}\|_F^2 + \lambda_1 \|\mathbf{Y}\|_1 \tag{2}$$

where $\|\mathbf{X} - \mathbf{D}\mathbf{Y}\|_F^2$ is the reconstruction error term evaluating how well a linear combination of the atoms (columns) of the dictionary $\mathbf{D}$, can approximate the data matrix $\mathbf{X}$, and $\|.\|_F$ denotes the matrix Frobenious norm. A sparsity regularisation term on the sparse code matrix $\mathbf{Y} \in \mathbb{R}^{d \times N}$, $\|\mathbf{Y}\|_1$ is added with a weighting factor λ_1 to favour a small number of atoms to be used for the reconstruction. When $d < r$, the original data $\mathbf{X}$ is represented by $\mathbf{Y}$ in a lower-dimensional subspace. Sparse coding is thus a dimensionality reduction method aiming to learn a more compact and descriptive representation of the data distribution so that the data latent structure and associations among data points can be more readily revealed by a clustering algorithm.

Assuming that the data reside on a low-dimensional manifold regularised by some unknown geometrical structure in the data, a graph regularised sparse coding model [26] can be formulated as:

$$\min_{\mathbf{D},\mathbf{Y}} \|\mathbf{X} - \mathbf{D}\mathbf{Y}\|_F^2 + \lambda_1 \|\mathbf{Y}\|_1 + \lambda_2 \Omega(\mathbf{Y}) \tag{3}$$

where $\Omega(\mathbf{Y})$ is the conventional graph regularisation term defined in Eq. (1) and the weight λ_2 controls its strength. Typically a k-nearest-neighbours graph $\mathbf{G}$ is constructed using $\mathbf{X}$ and the affinity matrix $\mathbf{W}$ as defined earlier is computed as either binary or using a Gaussian heat kernel on this graph. As analysed previously, this $\Omega(\mathbf{Y})$ graph regularisation term has two problems: (1) using a squared l_2-norm leads to sensitivity to outlying samples; and (2) constructing the graph $\mathbf{G}$ using $\mathbf{X}$ is suboptimal. To address these problems, we explore the following two considerations.

Robust graph regularisation (RGR). To solve the first problem, we propose a robust graph regularisation term, which has a graph weighted l_1-norm. More specifically, Eq. (1) is first rewritten in a matrix-form using trace notation:

$$\Omega(\mathbf{Y}) = \sum_{ij}^{N} \mathbf{W}_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 = tr(\mathbf{Y}\mathbf{L}_\mathbf{W}\mathbf{Y}^\mathbf{T}), \tag{4}$$

where $\mathbf{L}_\mathbf{W} = \mathbf{D} - \mathbf{W}$ is the Laplacian matrix, $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$ is a degree matrix. Let $\mathbf{L}_\mathbf{W} = \mathbf{U}_\mathbf{W}\mathbf{S}_\mathbf{W}\mathbf{U}_\mathbf{W}^\mathbf{T}$ using the eigen decomposition technique, and after some matrix manipulation, we have

$$tr(\mathbf{Y}\mathbf{L}_\mathbf{W}\mathbf{Y}^\mathbf{T}) = tr(\mathbf{Y}\mathbf{U}_\mathbf{W}\mathbf{S}_\mathbf{W}\mathbf{U}_\mathbf{W}^\mathbf{T}\mathbf{Y}^\mathbf{T}) = tr(\mathbf{Y}\mathbf{U}_\mathbf{W}\mathbf{S}_\mathbf{W}^{\frac{1}{2}}\mathbf{S}_\mathbf{W}^{\frac{1}{2}}\mathbf{U}_\mathbf{W}^\mathbf{T}\mathbf{Y}^\mathbf{T}) = \|\mathbf{Y}\mathbf{A}_\mathbf{W}\|_F^2, \tag{5}$$

where $\mathbf{A}_\mathbf{W} = \mathbf{U}_\mathbf{W}\mathbf{S}_\mathbf{W}^{\frac{1}{2}}$. Eq. (5) above is quadratic. To promote sparsity, we propose to use l_1-norm instead of Frobenius norm. That gives the proposed graph weighted l_1-norm regularisation term

$$\Omega_{R1}(\mathbf{Y}) = \|\mathbf{Y}\mathbf{A}_\mathbf{W}\|_1. \tag{6}$$

Replacing $\Omega(\mathbf{Y})$ with $\Omega_{R1}(\mathbf{Y})$ in Eq. (3), we have a robust graph regularised dictionary learning model.

Learning Graph. To solve the second problem, instead of computing $\mathbf{W}$ using $\mathbf{X}$, we assume that $\mathbf{W}$ (hence the graph $\mathbf{G}$ as $\mathbf{W}$ depends on the topology of $\mathbf{G}$) is unknown and has to be learned together with $\mathbf{D}$ and $\mathbf{Y}$. In order to learn $\mathbf{W}$, inspired by [10], a graph learning algorithm is developed and integrated into the RGR dictionary learning model proposed in this paper.

Now with these two changes to the graph regularisation term, our dictionary learning model with a learned RGR term can be formulated as:

$$\begin{aligned} \min_{\mathbf{D},\mathbf{Y},\mathbf{W}} \quad & \frac{1}{2}\|\mathbf{X} - \mathbf{D}\mathbf{Y}\|_F^2 + \lambda_1\|\mathbf{Y}\|_1 + \lambda_2\|\mathbf{Y}\mathbf{A}_\mathbf{W}\|_1 + \lambda_3\|\mathbf{W}\|_F^2 \\ \text{s.t.} \quad & \|\mathbf{d}_i\|^2 \leq 1, \mathbf{W}^\mathbf{T}\mathbf{1} = 1, \mathbf{W} \geq 0. \end{aligned} \tag{7}$$

where $\lambda_3\|\mathbf{W}\|_F^2$ is a regularisation term on $\mathbf{W}$ to prevent trivial solutions. λ_1, λ_2, and λ_3 control the contributions of the three regularisation terms, respectively. The constraints, $\mathbf{W}^\mathbf{T}\mathbf{1} = 1$ and $\mathbf{W} \geq 0$, are there to ensure the validity of the learned graph, while the constraint $\|\mathbf{d}_i\|^2 \leq 1$ ($\mathbf{d}_i$ is a column of $\mathbf{D}$ with $i = 1, ..., r$) enforces the learned dictionary atoms to be compact.

2.2 Optimisation

The problem in Eq. (7) is non-convex and non-smooth. Solving them is thus more difficult than both (2) and (3) due to the additional unknown variable $\mathbf{W}$ and the l_1-norm used in $\Omega_{R1}(\mathbf{Y})$. Next, we develop an efficient solver for Eq. (7) based on the Alternating Direction Method of Multipliers (ADMM) [2].

First, we transform the Eq. (7) by letting $\mathbf{S} = \mathbf{Y}$ and $\mathbf{U} = \mathbf{S}\mathbf{A}_\mathbf{W}$, then the Augmented Lagrangian function of Eq. (7) with the two introduced constraints is:

$$\begin{aligned}\mathcal{L}_{(\mathbf{D},\mathbf{Y},\mathbf{S},\mathbf{U},\mathbf{W})} =& \frac{1}{2}\|\mathbf{X}-\mathbf{D}\mathbf{S}\|_F^2 + \lambda_1\|\mathbf{Y}\|_1 + \lambda_2\|\mathbf{U}\|_1 \\ &+ \lambda_3\|\mathbf{W}\|_F^2 + \langle \mathbf{G}, \mathbf{Y}-\mathbf{S}\rangle + \frac{\gamma}{2}\|\mathbf{Y}-\mathbf{S}\|_F^2 \\ &+ \langle \mathbf{F}, \mathbf{U}-\mathbf{S}\mathbf{A}_\mathbf{W}\rangle + \frac{\gamma}{2}\|\mathbf{U}-\mathbf{S}\mathbf{A}_\mathbf{W}\|_F^2 \\ s.t. \quad & \|\mathbf{d}_i\|^2 \leq 1, \mathbf{W}^\mathbf{T}\mathbf{1} = 1, \mathbf{W} \geq 0.\end{aligned} \tag{8}$$

where, $\mathbf{F}$ and $\mathbf{G}$ are Lagrangian multipliers, and γ is a penalty parameter. Now, we can solve it in an alternating manner. To this end, we divide the Eq. (8) into three main steps (second and third step consists of a few substeps): First step is for learning $\mathbf{D}$ {Step 1}, second step is for learning $\mathbf{Y}$ {Step 2.1}, and updating auxiliary variables $\mathbf{S}$ {Step 2.2}, $\mathbf{U}$ {Step 2.3}, $\mathbf{F}$ and $\mathbf{G}$ {Step 2.4}, and third step is for learning graph $\mathbf{W}$ {Step 3}.

Step 1) Solving for D: When we learn $\mathbf{D}$ for the given $\mathbf{S}$ the objective function reduces to:

$$\min_{\mathbf{D}} \frac{1}{2}\|\mathbf{X}-\mathbf{D}\mathbf{S}\|_F^2 \quad s.t. \;\; \|\mathbf{d}_i\|^2 \leq 1, \;\; i = 1, ..., r \tag{9}$$

To solve Eq. (9), we use the Lagrange dual method as in [14]. The analytical solution of $\mathbf{D}$ can be computed as: $\mathbf{D}^* = \mathbf{X}\mathbf{S}^\mathbf{T}(\mathbf{S}\mathbf{S}^\mathbf{T} + \eta)^{-1}$, where η is a diagonal matrix constructed from all the optimal dual variables.

Step 2.1) Solving for Y: For the given $\mathbf{S}$, we solve the following objective to estimate $\mathbf{Y}$:

$$\min_{\mathbf{Y}} \lambda_1\|\mathbf{Y}\|_1 + \frac{\gamma}{2}\|\mathbf{Y} - (\mathbf{S} - \frac{\mathbf{G}}{\gamma})\|_F^2.$$

We can use the soft-thresholding operator to get $\mathbf{Y}$ as follows:

$$\mathbf{Y} = \text{sign}\left(\mathbf{S} - \frac{\mathbf{G}}{\gamma}\right)\max(\left|\mathbf{S} - \frac{\mathbf{G}}{\gamma}\right| - \frac{\lambda_1}{\gamma}). \tag{10}$$

Step 2.2) Solving for S: For the given $\mathbf{D}$, $\mathbf{Y}$, $\mathbf{W}$, and $\mathbf{U}$, we solve the following objective to estimate $\mathbf{S}$:

$$\min_{\mathbf{S}} \frac{1}{2}\|\mathbf{X}-\mathbf{D}\mathbf{S}\|_F^2 + \langle \mathbf{G}, \mathbf{Y}-\mathbf{S}\rangle + \frac{\gamma}{2}\|\mathbf{Y}-\mathbf{S}\|_F^2 + \langle \mathbf{F}, \mathbf{U}-\mathbf{S}\mathbf{A}_\mathbf{W}\rangle + \frac{\gamma}{2}\|\mathbf{U}-\mathbf{S}\mathbf{A}_\mathbf{W}\|_F^2.$$

Since every part of equation is quadratic, we can take derivative and set it zero which gives

$$(\mathbf{D}^\mathbf{T}\mathbf{D} + \gamma\mathbf{I})\mathbf{S} + \gamma\mathbf{S}\mathbf{A}_\mathbf{W}\mathbf{A}_\mathbf{W}^\mathbf{T} = \mathbf{D}^\mathbf{T}\mathbf{X} + \gamma\mathbf{U}\mathbf{A}_\mathbf{W}^\mathbf{T} + \gamma\mathbf{Y} + \mathbf{G} + \mathbf{F}\mathbf{A}_\mathbf{W}^\mathbf{T}. \tag{11}$$

This is a well-known Sylvester equation that can be solved by Bartels-Stewart algorithm [1].
Step 2.3) Solving for U: For the given $\mathbf{S}$, we solve the following objective to estimate $\mathbf{U}$:

$$\min_{\mathbf{U}} \lambda_2\|\mathbf{U}\|_1 + \frac{\gamma}{2}\|\mathbf{U} - (\mathbf{S}\mathbf{A}_\mathbf{W} - \frac{\mathbf{F}}{\gamma})\|_F^2.$$

Again, we can use the soft-thresholding operator to get $\mathbf{U}$ as Step **2**.

Step 2.4) Updating Multipliers: $\mathbf{G}, \mathbf{F}, \gamma$ (as common practise, $\mathbf{G}$ and $\mathbf{F}$ are initiliased by zero matrix, and $\gamma = 0.01$).

$$\mathbf{G} = \mathbf{G}^{old} + \gamma(\mathbf{Y} - \mathbf{S}), \quad \mathbf{F} = \mathbf{F}^{old} + \gamma(\mathbf{U} - \mathbf{S}\mathbf{A}_{\mathbf{W}}), \quad \gamma = \rho\gamma^{old}$$

Step 3) Solving for W: After learning $\mathbf{Y}$ from previous step (Step 2), we have a following formulation to learn $\mathbf{W}$:

$$\min_{\mathbf{W}} \quad \lambda_2 \sum_{ij}^{N} \mathbf{W}_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|_1 + \lambda_3 \|\mathbf{W}\|_F^2 \;\; s.t. \;\; \mathbf{W}_i^T \mathbf{1} = 1, \mathbf{W}_i \geq 0 \tag{12}$$

Note that we put λ_2 to control the effect of the regularisation as in Eq. (7) [1]. To solve Eq. 12, first let's denote $\mathbf{d}_{ij} = \frac{\lambda_2 \|\mathbf{y}_i - \mathbf{y}_j\|_1}{2\lambda_3}$ and $\|\mathbf{W}\|_F^2 = \sum_{ij} \mathbf{W}_{ij}^2$, then we can have a closed-form solution by using Lagrange multipliers [10, 20] for this problem:

$$\mathbf{W}_i = \left(\frac{1 + \sum_{j=1}^{k} \tilde{\mathbf{d}}_j}{k} \mathbf{1} - \mathbf{d}_i \right)_+ \tag{13}$$

where the operator $(\mathbf{q})_+$ projects negative elements in $\mathbf{q}$ to 0. $\tilde{\mathbf{d}}_i$ is $\mathbf{d}_i$ but with ascending order. After obtaining $\mathbf{W}$ we symmetrise it using $\frac{\mathbf{W}+\mathbf{W}^\mathbf{T}}{2}$. Then, we build Laplacian matrix and decompose it according to Eq. (5) and Eq. (6).
We continue to alternate solving for $\mathbf{D}$, $\mathbf{Y}$, $\mathbf{S}$, $\mathbf{U}$, $\mathbf{W}$ until a predefined threshold (10^{-3}) is satisfied. With the learned estimated sparse code $\mathbf{Y}$ as the new data representation and affinity matrix $\mathbf{W}$, the final clustering result is obtained using the spectral clustering algorithm in [21] as done by most subspace clustering methods.

Convergence Analysis. In practice, it is guaranteed that the objective function converges to a local stable point [2], although the theoretical convergence proof of ADMM for global optimum does not exist. This is validated by our experiments (see Sec. 3). Also, see the supplementary material (Sec. A) for the complexity analysis.

3 Experiments

3.1 Datasets and Settings

Table 1: Dataset summary

Datasets	#Samples	#Classes	Type/Challenges
C-PIE	1428	68	Face/Illumination
ORL	400	40	Face/Pose
YaleB	2414	38	Face/Illumination/Pose
Yale	165	15	Face/Illumination
COIL	1440	20	Object/Pose

Datasets. Most subspace clustering methods were tested on image clustering tasks because of the high dimensionality. Five most widely used image clustering benchmarks are chosen: CMU-PIE (C-PIE for short)[2], ORL[3], YaleB[4], Yale[5], and COIL20 (COIL for short)[6]. Among

[1] Note that $\sum_{ij}^{N} \mathbf{W}_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|_1$ is an approximation for $\|\mathbf{A}_\mathbf{W}\mathbf{Y}\|_1$ when solving the objective with respect to $\mathbf{W}$.
[2] http://vasc.ri.cmu.edu/idb/html/face/
[3] http://www.cl.cam.ac.uk/research/dtg/attarchive/projects.html
[4] http://vision.ucsd.edu/ leekc/ExtYaleDatabase/ExtYaleB.html
[5] http://vision.ucsd.edu/content/yale-face-database
[6] http://www.cs.columbia.edu/CAVE/software/softlib/coil20.php

them, four are complex face datasets with pose/illumination/facial expression changes, and the last one (COIL) contains different general objects with pose changes. A summary of the datasets is given in Table 1.

Settings. In all experiments, we follow the standard settings used in most existing works. Specifically, in all the datasets, the size of image is resized to 32×32 and each image is represented by a 1024-dimensional vector by concatenating the pixel intensity values. For C-PIE, we fix the pose and expression; thus there are 21 images for each subject under different illumination conditions. For YaleB, only the images of the first 10 subjects are used as in [10, 11, 17]. The number of clusters is given as the true class number as the other works do. For those two-stepped matrix factorisation based methods, after dimensionality reduction in the first step, we use the Normalised Cut [21] spectral clustering algorithm to produce the final clustering results. Similar to other graph regularised subspace learning models, there are a number of free parameters in our model. As the problem is unsupervised clustering, they cannot be cross-validated. We do not tune them for individual dataset. Instead, a fixed set of values are set to them for all datasets. Specifically, we use 256 atoms (i.e. $d = 256$) for the compared dictionary learning based methods including ours; other parameters for our model are: $k = 5$ in the k-nearest-neighbour graph for computing $\mathbf{W}$; different weighting factors λ_1, λ_2 and λ_3 (see Eq. (7)) are tuned among the values of $10^{\{-4,\ldots,2\}}$ as a common practice. A sensitivity analysis on these parameters is presented later. The implementation of our method and demo code can be downloaded from the first author's homepage.

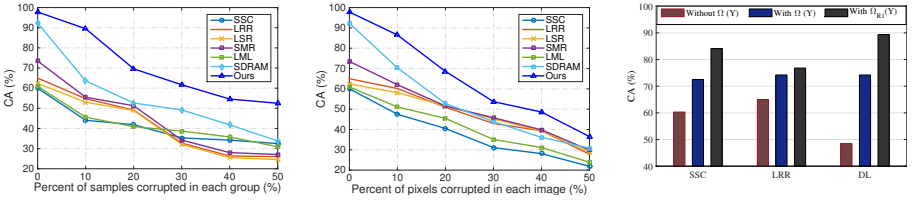
Evaluation Metrics. Two standard clustering metrics are used for measuring the clustering performance: clustering accuracy (CA) and normalised mutual information metric (NMI) [4]. For each dataset, we run experiments 50 times, and average results are reported.

Competitors. We select 12 recently published state-of-the-art methods for comparison with a focus on choosing the most related and competitive methods. They can be categorised into three groups depending on whether their objective functions contain a graph regularisation term and whether graph learning is deployed. *Five non-graph-based methods:* l_1-graph (l_1G) [6] – essentially dictionary learning for sparse coding (Eq. (2)); SSC – self-expressive model with l_1-norm regularisation [9]; LRR – self-expressive model with nuclear-norm regularisation [15]; LSR – self-expressive model with Frobenius regularisation [16]; and CASS – self-expressive model with trace-norm regularisation [17]. *Four graph-based methods:* GSC – [4] dictionary learning for sparse coding with the conventional graph regularisation term (Eq. (1)); SMR – self-expressive model with conventional graph regularisation [11]; Rl_1G – self-expressive model with iterative updating of the graph using $\mathbf{Y}$ [24]; NSLRR – SMR+LRR but with hypergraph for robustness [25]. *Three graph-learning-based methods:* PCAN – LPP with graph learning method that uses the conventional graph regularisation term [20]; SDRAM – self-expressive model with graph learning on the conventional graph regularisation term [10]; LML [8] – matrix factorisation model and again with graph learning but with the graph regularisation term in Eq. (1).

These compared methods therefore cover a wide ranges of clustering methods from sparse subspace, matrix factorisation, to dictionary learning, with a number of them (e.g. NSLRR, SDRAM, LML) proposed quite recently. They are thus representative of the state-of-the-art. All methods have their source code available so that they can be evaluated under exactly the same setting, with the only exception of NSLRR, for which the reported results in [25] are presented. For all compared methods, the parameters are tuned according to their respective papers and the best performance is reported.

Table 2: Comparative results on C-PIE, COIL, ORL, Yale, and YaleB. 'G' stands for graph.

		CA (%)					NMI (%)				
Methods	**G**	**C-PIE**	**COIL**	**ORL**	**Yale**	**YaleB**	**C-PIE**	**COIL**	**ORL**	**Yale**	**YaleB**
l_1G [6]	No	70.3	67.1	66.8	40.0	48.4	88.3	77.3	76.5	46.1	51.4
SSC [9]	No	72.1	58.9	55.5	38.7	60.3	80.0	68.6	66.5	36.0	52.6
LRR [15]	No	71.5	45.1	66.2	46.2	65.0	90.3	61.6	78.7	50.1	66.2
LSR [16]	No	77.9	56.0	56.5	48.5	62.4	91.9	61.5	67.6	39.7	65.7
CASS [17]	No	82.6	59.1	68.3	45.6	81.9	92.3	64.1	78.1	52.2	78.1
GSC [22]	Yes	**100**	80.9	61.5	43.4	74.2	**100**	87.5	76.2	46.9	75.0
R/l_1G [22]	Yes	89.5	79.4	62.0	41.3	68.5	90.2	79.6	78.4	45.9	69.4
SMR [10]	Yes	85.4	65.6	57.6	45.3	73.5	93.2	74.5	61.4	57.5	76.7
NSLRR [23]	Yes	85.1	61.8	55.3	NA	NA	96.6	75.6	74.5	NA	NA
PCAN [20]	Yes	82.5	76.5	49.1	54.4	59.2	85.6	79.6	53.7	54.7	60.3
LML [8]	Yes	90.2	80.2	46.7	46.7	60.9	96.3	85.3	52.3	52.3	65.2
SDRAM [11]	Yes	95.6	86.3	70.6	51.8	92.3	98.8	89.1	80.2	56.1	89.1
Ours	Yes	**100**	**88.1**	**76.3**	**59.6**	**95.2**	**100**	**89.3**	**86.1**	**60.6**	**94.2**



(a) Type I noise (b) Type II noise (c) Effects of RGR

Figure 1: Further results on YaleB with additional Type I noise (a) and Type II noise (b), averaged over 50 trials and (c) using three existing subspace learning models extended by using our robust l_1-norm graph regularisation term.

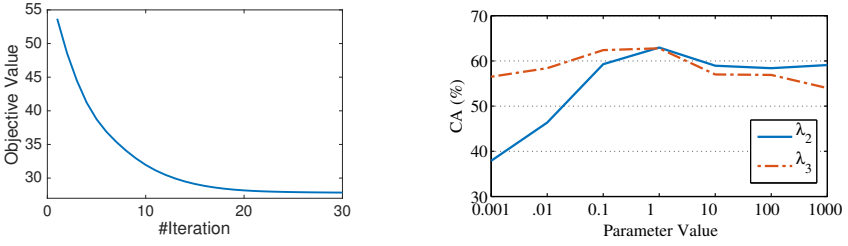
3.2 Comparative Results

Comparative results are shown in Table 2. We can make the following observations: (1) Our method achieves the best performance on all five datasets with both metrics. On the easiest dataset, C-PIE (no pose changes, moderate illumination changes), ours is tied with GSC with perfect clustering. On the other four datasets, there is a clear margin between ours and that of the second best. (2) Overall, the methods with graph regularisation outperform the methods without. (3) Many methods, including very recent ones such as LML and SMR struggled with YaleB which features dramatic illumination changes. Yet, our method copes with this condition very well indicating that the outlying samples caused by extreme illumination condition can be dealt with effectively by our learned graph regularisation term with robust l_1-norm. (4) The contribution of our learned RGR is most distinctive when comparing ours with GSC: both are graph regularised sparse coding models with the only difference being that our term is learned and with the l_1-norm. The performance gain measured in clustering accuracy achieved by our model is as big as 21% (YaleB). (5) The recently proposed three graph-learning-based methods (PCAN, LML and SDRAM) are competitive. However, our model is clearly superior to them thanks to the robust l_1-norm adopted in our formulation.

Robustness against Noise. In this experiment, we replace random pixels in images with noise to create additional outliers in order to further evaluate the robustness of the compared methods. Two types of noise are introduced to the YaleB dataset. For Type I, different percentage (from 0% to 50%) of the samples are randomly selected to add 50% pixel-level noise, that is, by replacing randomly 50% of the image pixels with values following a uniform dis-

Table 3: Running time comparison on YaleB

Method	SSC	LRR	LSR	CASS	LML	SDRAM	GSC	l_1G	PCAN	SMR	Ours
Time (s)	70	90	0.02	35,560	28,660	50	1,160	770	50	1.5	40



(a) (b)

Figure 2: (a) Convergence curve (YaleB) and (b) Parameter sensitivity (COIL).

tribution on the interval from 0 to 255. For Type II, each and every image is corrupted, and the percentage of corrupted pixels per image varies from 0 % to 50%.

Note that most existing works on robust clustering evaluated on the Type II noise. However, in a real-world scenario, it is unlikely that all data are corrupted by noise by the same degree. In contrast, the Type I noise creates variable numbers of outlying samples and has an uneven effect on different samples. It is thus more realistic. Our method is compared with six representative competitors and the results are shown in Figure 1. It can be seen in Figure 1(a) that with more outlying samples created by adding the Type I noise, the gaps between our method and the competitors get bigger. This result suggests that our method is more robust than the competitors particularly when more of the realistic Type I noisy samples are present. A different trend is observed in Figure 1(b) with the Type II noise, where the gaps seems to be stable given different amounts of noise. Some qualitative results are shown in the supplementary material (Sec. C).

Effects of Robust Graph Regularisation. In this experiment, we look at three subspace learning models including SSC, LRR, and Dictionary Learning (DL, i.e. l_1G) and examine how they can benefit from graph regularisation in general and our robust l_1-norm graph regularisation in particular. The results in Figure 1(c) shows that: (1) all three models benefit from graph regularisation and (2) the performance gain is larger with our robust l_1-norm graph regularisation term compared with the conventional term in Eq. (1). These results thus suggest that the proposed RGR term is generally applicable to models beyond dictionary learning or even subspace learning and unsupervised clustering.

Convergence Analysis. Figure 2(a) shows the convergence of the objective function (Eq. (7)) using the developed optimisation algorithm on the COIL dataset. It can be seen that the objective function value decreases rapidly and converges after 15 iterations. A similar trend is observed for all other datasets (see the supplementary Sec. A).

Running Time Comparison. Table 3 compares the running times of different methods, obtained on a PC with 3.40GHz CPU, 16GB memory and MATLAB implementation. It shows that our model (Ours) is very competitive in terms of computational efficiency, although models without any l_1-norm term such as LSR and SMR are naturally more efficient.

Sensitivity on Free Parameters and Individual Components of Our Model. Figure 2(b) shows the effects of varying the values of λ_2 or λ_3 (Eq. 7) whilst fixing the values of other parameters. We note that our method is in general not sensitive to λ_3. The effect of λ_2 is relatively small when $\lambda_2 > 1$. However, when $\lambda_2 < 1$, smaller values of λ_2 leads to weaker

performance because the contribution of the graph regularisation term diminishes. As for λ_1, we find that as long as it is small, its effect is neglectable. We thus fixed $\lambda = 0.001$ in all experiments. The same is observed for the value of k in the k-nearest-neighbour graph. Its value is fixed to 5 throughout. Furthermore, we analysed the sensitivity of dictionary size which is not sensitive (see the supplementary material (Sec. B)).

4 Conclusion

We have proposed a novel robust clustering algorithm based on l_1-norm graph regularised dictionary learning for sparse coding. Compared with existing graph regularised matrix factorisation/subspace learning based clustering methods, our method uses l_1-norm regularisation that is more robust against outliers, and learns optimal graph weights jointly with the subspace learning rather than fixing the graph in the low-level feature space. Our extensive experiments have validated the effectiveness and efficiency of the proposed method.

Acknowledgements

The authors were funded in part by the European Research Council under the FP7 Project SUNNY (grant agreement no. 313243).

References

[1] Richard H. Bartels and GW Stewart. Solution of the matrix equation ax+ xb= c [f4]. *Communications of the ACM*, 15(9):820–826, 1972.

[2] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[3] Paul S Bradley and Olvi L Mangasarian. k-plane clustering. *Journal of Global Optimization*, 16(1):23–32, 2000.

[4] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1548–1560, 2011.

[5] Bin Cheng, Jianchao Yang, Shuicheng Yan, Yun Fu, and Thomas S Huang. Learning with l_1-graph for image analysis. *IEEE Transactions on Image Processing*, 19(4):858–866, 2010.

[6] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

[7] Samuel I Daitch, Jonathan A Kelner, and Daniel A Spielman. Fitting a graph to vector data. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 201–208. ACM, 2009.

[8] Xiaowen Dong, Dorina Thanou, Pascal Frossard, and Pierre Vandergheynst. Laplacian matrix learning for smooth graph signal representation. In *Proceedings of IEEE ICASSP*, number EPFL-CONF-205027, 2015.

[9] Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2790–2797. 2009.

[10] Xiaojie Guo. Robust subspace segmentation by simultaneously learning data representations and their affinity matrix. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[11] Han Hu, Zhouchen Lin, Jianjiang Feng, and Jie Zhou. Smooth representation clustering. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3834–3841. IEEE, 2014.

[12] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[13] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS*, 2001.

[14] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2006.

[15] Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 663–670, 2010.

[16] Canyi Lu, Hai Min, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan. Robust and efficient subspace segmentation via least squares regression. In *Computer Vision–ECCV 2012*, pages 347–360. Springer, 2012.

[17] Canyi Lu, Jiashi Feng, Zhouchen Lin, and Shuicheng Yan. Correlation adaptive subspace segmentation by trace lasso. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1345–1352. IEEE, 2013.

[18] Yi Ma, Harm Derksen, Wei Hong, and John Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(9):1546–1562, 2007.

[19] Yi Ma, Allen Y Yang, Harm Derksen, and Robert Fossum. Estimation of subspace arrangements with applications in modeling and segmenting mixed data. *SIAM review*, 50(3):413–458, 2008.

[20] Feiping Nie, Xiaoqian Wang, and Heng Huang. Clustering and projected clustering with adaptive neighbors. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 977–986. ACM, 2014.

[21] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[22] Pablo Sprechmann and Guillermo Sapiro. Dictionary learning and sparse coding for unsupervised clustering. In *ICASSP*, 2010.

[23] René Vidal. A tutorial on subspace clustering. *IEEE Signal Processing Magazine*, 28 (2):52–68, 2010.

[24] Yingzhen Yang, Zhangyang Wang, Jianchao Yang, Jiawei Han, and Thomas Huang. Regularized l1-graph for data clustering. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[25] Ming Yin, Junbin Gao, and Zhouchen Lin. Laplacian regularized low-rank representation and its applications. 2015.

[26] Miao Zheng, Jiajun Bu, Chun Chen, Can Wang, Lijun Zhang, Guang Qiu, and Deng Cai. Graph regularized sparse coding for image representation. *Image Processing, IEEE Transactions on*, 20(5):1327–1336, 2011.