Data-Dependent Sparsity for Subspace Clustering

Bo Xin
Microsoft Research, Beijing

Yizhou Wang
Peking University

Wen Gao
Peking University

David Wipf
Microsoft Research, Beijing

Abstract

Subspace clustering is the process of assigning subspace memberships to a set of unlabeled data points assumed to have been drawn from the union of an unknown number of low-dimensional subspaces, possibly interlaced with outliers or other data corruptions. By exploiting the fact that each inlier point has a sparse representation with respect to a dictionary formed by all the other points, an ℓ_1 regularized sparse subspace clustering (SSC) method has recently shown state-of-the-art robustness and practical extensibility in a variety of applications. But there remain important lingering weaknesses. In particular, the ℓ_1 norm solution is highly sensitive, often in a detrimental direction, to the very types of data structures that motivate interest in subspace clustering to begin with, sometimes leading to poor segmentation accuracy. However, as an alternative source of sparsity, we argue that a certain data-dependent, non-convex penalty function can compensate for dictionary structure in a way that is especially germane to subspace clustering problems. For example, we demonstrate that this proposal displays a form of invariance to feature-space transformations and affine translations that commonly disrupt existing methods, and moreover, in important settings we reveal that its performance quality is lower bounded by the ℓ_1 solution. Finally, we provide empirical comparisons on popular benchmarks that corroborate our theoretical findings and demonstrate superior performance when compared to recent state-of-the-art models.

1 INTRODUCTION

Subspace clustering is the process of segmenting a set of unlabeled data points that were drawn from the union of an unknown number of low-dimensional subspaces, possibly corrupted with outliers. As a generalization of traditional PCA, and a fundamental tool for data analysis in high dimensional settings, subspace clustering is relevant to numerous practical applications, including image representation and compression, motion segmentation, and face clustering (Elhamifar and Vidal, 2009; Soltanolkotabi and Candes, 2012; Elhamifar and Vidal, 2013; Liu et al., 2010; Rao et al., 2010; Lu et al., 2012; Liu et al., 2013; Feng et al., 2014; Lu et al., 2013; Soltanolkotabi et al., 2014). We define this problem more formally as follows.

Definition 1 *Let $\{\mathcal{S}_k\}_{k=1}^m$ denote a set of m linear (or possibly affine) subspaces in $\mathbb{R}^d$, where dim$[\mathcal{S}_k] = d_k < d \quad \forall k = 1, \ldots, m$. Moreover, suppose we have drawn n_k points from each subspace forming data matrices $\boldsymbol{X}_k \in \mathbb{R}^{d \times n_k}$. We then concatenate the points from each subspace and combine them with a matrix $\boldsymbol{X}_0 \in \mathbb{R}^{d \times n_0}$ whose columns represent outlying points with no subspace membership. Finally, the full arrangement of $n = n_0 + \sum_{k=1}^m n_k$ points is scrambled with an unknown permutation matrix $\boldsymbol{P} \in \mathbb{R}^{n \times n}$. The entire ensemble can then be expressed as*

$$\boldsymbol{X} \triangleq [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] = [\boldsymbol{X}_0, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_K]\boldsymbol{P} \in \mathbb{R}^{d \times n}. \tag{1}$$

Subspace clustering *is defined as the process of estimating the subspace membership of each point $\boldsymbol{x}_i$ while discarding the outliers, without any a prior knowledge of the dimensionality or cardinality of the underlying union of subspaces.*

Given some means of first isolating and removing outliers, spectral clustering represents one of the most robust approaches to obtaining accurate data segmentations. This process involves forming an affinity matrix $\boldsymbol{A}$, where the ij-th element a_{ij} quantifies the strength of the relationship between points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. In traditional clustering, this affinity is typically computed using a Gaussian kernel $\exp[-\alpha\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2]$ with $\alpha > 0$, but

this ignores the subspace structure we want to reflect.

To this end, we instead form $\boldsymbol{A}$ to honor the desired subspace arrangement by exploiting a *self-expressiveness property* of $\boldsymbol{X}$ (Elhamifar and Vidal, 2013), namely that any $\boldsymbol{x}_i$ can be represented as a linear combination of other data points in $\boldsymbol{X}$ within the same subspace. This of course assumes a suitable sampling of points in $\boldsymbol{X}$, i.e., each n_k is sufficiently large with points in general position. Overall, with $n > d$ there will exist an infinite number of such self-expressive representations; however, in constructing a viable affinity matrix it is paramount that we find representations that heavily favor *only* using points from the same subspace.

For the moment, assume that no outliers are present (i.e., $n_0 = 0$). At a motivational level, sparse subspace clustering (SSC) (Elhamifar and Vidal, 2013) attempts to learn subspace-aware representations by solving

$$\min_{\boldsymbol{Z}} \|\boldsymbol{Z}\|_0 \text{ s.t. } \boldsymbol{X} = \boldsymbol{X}\boldsymbol{Z}, \text{ diag}[\boldsymbol{Z}] = 0, \qquad (2)$$

where $\|\boldsymbol{Z}\|_0$ is the matrix ℓ_0 norm, or a count of the number of nonzero elements in $\boldsymbol{Z}$, a penalty function that heavily favors zero-valued elements or a sparse $\boldsymbol{Z}$. The diagonal constraint is included to prevent each point from using itself in the representation (e.g., the degenerate solution $\boldsymbol{Z}^* = \boldsymbol{I}$), ideally deferring to others in the same subspace. Provided that each individual subspace satisfies $d_k < d$ for all k, and sampled points are sufficiently dense in general position, then up to a permutation matrix $\boldsymbol{P}$, the solution to (2) will be block diagonal and aligned with the true clusters revealing subspace memberships. Note that if noise or other modeling errors are present the equality constraint $\boldsymbol{X} = \boldsymbol{X}\boldsymbol{Z}$ can be relaxed with the inclusion of an additional trade-off parameter, e.g., $\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z}\|_{\mathcal{F}} \leq \epsilon$.

From a practical standpoint, theoretical analysis from (Elhamifar and Vidal, 2013; Soltanolkotabi and Candes, 2012) suggests that in certain cases as long as the angles between subspaces are not too small and points within are suitably arranged, then we can replace the intractable, NP-hard matrix ℓ_0-norm minimization in (2) with $\|\boldsymbol{Z}\|_1$ and still compute the same desirable block-diagonal structure. We will henceforth refer to this algorithm as ℓ_1-SSC. If the data $\boldsymbol{X}$ ideally follow the union of subspace model and the ℓ_1-SSC solution matches the solution of (2), then it is possible to extract subspace memberships directly from $\boldsymbol{Z}^*$. However, in practical situations with noise and model mismatch, it is highly beneficial to first form a symmetric affinity matrix as $\boldsymbol{A} = |\boldsymbol{Z}^*| + |\boldsymbol{Z}^*|^\top$ and then apply traditional spectral clustering (Luxburg, 2007) to the normalized Laplacian of $\boldsymbol{A}$ to obtain a more robust segmentation (Elhamifar and Vidal, 2013).

Regardless, the more the self-expressive representation obtained by ℓ_1-SSC reflects the subspace alignments and sizes, the more effective the final spectral clustering step will be. But is the ℓ_1 norm the optimal objective? Certainly a variety of surrogates for producing a block diagonal $\boldsymbol{Z}$ have been proposed in the literature such as the nuclear norm or other rank penalties (Babacan et al., 2012; Liu et al., 2013), the Frobenius norm (Lu et al., 2012), and the Trace Lasso (Lu et al., 2013). Likewise, a multi-task learning approach from (Wang et al., 2015) introduces an additional penalty and tuning paramter to encourage subspace-aware group-sparse representations. But from a theoretical standpoint, of the existing methods ℓ_1-SSC enjoys some of the strongest recovery guarantees (Soltanolkotabi and Candes, 2012), requires minimal tuning or a priori knowledge across broad operating conditions, and moreover, it is perhaps one of the most straightforward to adapt for handling practical situations where outliers or other data corruptions are present (Elhamifar and Vidal, 2013).

And yet there remains critical lingering issues with ℓ_1-SSC (issues that also affect many other algorithms in the literature). The equivalence between the ℓ_0 and ℓ_1 solutions will be quite sensitive to correlation structure in $\boldsymbol{X}$ as well as the particular feature representation used for each $\boldsymbol{x}_i$ and its interrelationship with the corresponding column norms. In brief, while the ℓ_1 norm regularizer is an order-wise optimal substitution for the ℓ_0 norm with iid Gaussian or similar designs (see (Candes et al., 2006) and the vast literature on compressive sensing), this relationship completely breaks down with the types of highly-structured data required by subspace clustering problems. And of course the reality is, *if such confounding structures were not present, there would be little value in attempting subspace clustering to begin with.*[1]

Fortunately though, unlike the ℓ_1 norm or other typical penalty functions, there exist data-dependent regularization techniques that can directly compensate for correlated data in finding maximally sparse or minimal ℓ_0-norm representations (Wipf, 2011), suggesting a seemingly natural surrogate for sparse subspace segmentation. In Section 2 we will introduce one specific, so-called *data-dependent* penalty for *sparse subspace clustering* (DD-SSC) procedure, as well as principled modifications to handle outliers and affine subspaces.

Note that nearly all recent subspace clustering work be-

[1] Technically speaking, it is sometimes possible to achieve a correct clustering even when ℓ_0-ℓ_1 norm equivalency does not hold (You and Vidal, 2015). However, it is still well-known that the ℓ_0 norm will provably produce a correct subspace clustering under *much* broader conditions than the ℓ_1 norm (e.g., see (Yang et al., 2016)).

gins with some existing algorithm or penalty function over $\boldsymbol{Z}$ that is then optimized to produce a viable affinity matrix. The novelty then lies in the attendant technical arguments for why a particular approach is likely to produce the desired, subspace-aligned block-diagonal structure, and ultimately the correct segmentation. In most cases (e.g.,(Liu et al., 2013; Lu et al., 2012, 2013)), theoretical recovery guarantees are restricted to the simplified setting where the subspaces are independent, in which case virtually any regularizer for $\boldsymbol{Z}$ is provably adequate. In contrast, much stronger guarantees have been shown for convex ℓ_1-SSC using rigorous geometric considerations (Elhamifar and Vidal, 2013; Soltanolkotabi and Candes, 2012; You and Vidal, 2015). Building upon these results, Section 3 presents our two primary analytical contributions, which can be summarized as follows:

- We demonstrate broad, challenging conditions under which DD-SSC will provably perform as well or better than ℓ_1-SSC in a sense that no existing algorithm can match.

- Several important factors can conspire to significantly disrupt the performance of current state-of-the-art subspace clustering algorithms, often in subtle underappreciated ways. These include the aggregate effects of feature space transformations (meaning invertible, potentially ill-conditioned transformations of the columns of $\boldsymbol{X}$), affine subspace translations, and the effects of disparate column norms $\|\boldsymbol{x}_i\|_2$. We precisely characterize the nature of these confounds and demonstrate that DD-SSC is largely invariant to all three factors, unlike existing approaches.

Finally, Section 4 provides complementary empirical results using both simulated and real-world data that confirm our main theoretical insights. Additionally, after preparing an initial version of our paper, we noticed an interesting recent work that also emphasizes the importance of minimizing the ℓ_0 norm as opposed to defaulting to the convex ℓ_1 relaxation (Yang et al., 2016). However, the proposed algorithmic strategy for actually accomplishing this, which amounts to regular ℓ_1 norm minimization followed by non-convex iterative hard-thresholding (IHT) iterations (Blumensath and Davies, 2008) to approximate the ℓ_0 norm, is radically different from ours. Importantly, it does not actually address any of the issues we raise above. In fact, this approach will be highly sensitive to feature space transformations and affine subspace translations since both algorithmic components, meaning the ℓ_1 and IHT steps, are heavily influenced by correlations and scaling factors in the data $\boldsymbol{X}$, and provably do not possess the desirable invariances of DD-SSC. Moreover, empirically we find that DD-SSC consistently outperforms the method from (Yang et al., 2016) on a battery of tests drawn from the latter.

2 DATA-DEPENDENT SPARSE SUBSPACE CLUSTERING (DD-SSC)

Model Development: The original sparse subspace clustering problem from (2) decouples into n individual NP-hard sparse estimation tasks of the form

$$\min_{\boldsymbol{z}_i} \|\boldsymbol{z}_i\|_0 \quad \text{s.t. } \boldsymbol{x}_i = \boldsymbol{X}_{\bar{i}}\boldsymbol{z}_i, \tag{3}$$

where $\boldsymbol{X}_{\bar{i}}$ denotes the full data matrix $\boldsymbol{X}$ with the i-th column removed.[2] Instead of the tractable convex substitution $\|\boldsymbol{z}_i\|_1$ adopted by ℓ_1-SSC, we advocate an alternative, data-dependent penalty that ultimately emerges from manipulations of a seemingly-unrelated Bayesian model from (Tipping, 2001) applied to each $\boldsymbol{x}_i$. For completeness, we briefly introduce the high-level derivations as follows.

For the i-th data point we first define the Gaussian likelihood

$$p(\boldsymbol{x}_i|\boldsymbol{z}_i;\boldsymbol{X}_{\bar{i}},\alpha) \propto \exp\left[-\tfrac{1}{2\alpha}\left\|\boldsymbol{X}_{\bar{i}}\boldsymbol{z}_i - \boldsymbol{x}_i\right\|_2^2\right]. \tag{4}$$

We also note that in the limit as $\alpha \rightarrow 0$, this likelihood will enforce the same constraint set as in (3) with probability one. For other values of α we may of course account for measurement noise or model mismatch errors as appropriate. Next we assume parameterized zero-mean Gaussian distributions as priors over each $\boldsymbol{z}_i$. Specifically, we have

$$p(\boldsymbol{z}_i;\boldsymbol{\gamma}_i) \propto \exp\left[-\tfrac{1}{2}\boldsymbol{z}_i^\top \boldsymbol{\Gamma}_i^{-1}\boldsymbol{z}_i\right], \quad \boldsymbol{\Gamma}_i \triangleq \text{diag}[\boldsymbol{\gamma}_i], \tag{5}$$

where $\boldsymbol{\gamma}_i$ denotes a vector of unknown variance hyperparamters. Given that both likelihood and prior are Gaussian, the posterior $p(\boldsymbol{z}_i|\boldsymbol{x}_i;\boldsymbol{X}_{\bar{i}},\boldsymbol{\gamma}_i,\alpha)$ is also Gaussian, with mean $\hat{\boldsymbol{z}}_i$ given by

$$\hat{\boldsymbol{z}}_i = \boldsymbol{\Gamma}_i\boldsymbol{X}_{\bar{i}}^\top\boldsymbol{\Sigma}_{\boldsymbol{x}_i}^{-1}\boldsymbol{x}_i, \quad \text{with} \tag{6}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{x}_i} \triangleq \boldsymbol{X}_{\bar{i}}\boldsymbol{\Gamma}_i\boldsymbol{X}_{\bar{i}}^\top + \alpha\boldsymbol{I}. \tag{7}$$

From the above expressions it is clear that if $\boldsymbol{\gamma}_i$ is a sparse vector with mostly zero-valued entries, then by virtue of its diagonal positioning and lefthand-side multiplication in (6), the estimator $\hat{\boldsymbol{z}}_i$ will have a matching sparsity profile or support. Of course for this framework to be a successful entry point for producing sparsity, we require some way of determining a viable estimate for $\boldsymbol{\gamma}_i$.

[2] Note that per this definition, the dimension of $\boldsymbol{z}_i$ will technically be reduced by one relative to the original columns of $\boldsymbol{Z}$ since we have removed the zero-valued elements enforced by the constraint; however, we retain this admittedly inconsistent notation for convenience.

If we temporarily treat the unknown $\boldsymbol{z}_i$ as a nuisance variable and $\boldsymbol{\gamma}_i$ as the parameter of interest, a typical empirical Bayesian estimation strategy is to marginalize over $\boldsymbol{z}_i$ and then maximize the resulting type-II likelihood function with respect to $\boldsymbol{\gamma}_i$ (MacKay, 1992). Fortunately, the resulting convolution of Gaussians integral is available in closed-form (Tipping, 2001) such that we can equivalently minimize the negative log-likelihood

$$\begin{aligned}\mathcal{L}(\boldsymbol{\gamma}_i) &= -\log \int p(\boldsymbol{x}_i|\boldsymbol{z}_i; \boldsymbol{X}_{\bar{i}}, \alpha) p(\boldsymbol{z}_i; \boldsymbol{\gamma}_i) d\boldsymbol{z}_i \\ &\equiv \boldsymbol{x}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{x}_i}^{-1} \boldsymbol{x}_i + \log |\boldsymbol{\Sigma}_{\boldsymbol{x}_i}|. \end{aligned} \tag{8}$$

Once we have an estimate of $\boldsymbol{\gamma}_i$ for all data points, we can compute a final estimator $\hat{\boldsymbol{Z}}$ by concatenating the respective posterior means computed via (6).

Thus far we have essentially just deferred a direct search for a sparse estimator of $\boldsymbol{z}_i$ to the search for a (hopefully) sparse $\boldsymbol{\gamma}_i$. However, if we apply (Wipf et al., 2011, Theorem 2), we can convert the estimation problem of minimizing $\mathcal{L}(\boldsymbol{\gamma}_i)$ in $\boldsymbol{\gamma}_i$-space to an equivalent problem in $\boldsymbol{z}_i$-space, facilitating direct analysis and comparison with more traditional sparse estimators. In particular, it can be shown that minimizing (8) and then computing $\hat{\boldsymbol{z}}_i$ using (6) is equivalent to minimizing

$$\mathcal{L}(\boldsymbol{z}_i) \triangleq \tfrac{1}{\alpha} \|\boldsymbol{x}_i - \boldsymbol{X}_{\bar{i}} \boldsymbol{z}_i\|_2^2 + f(\boldsymbol{z}_i; \boldsymbol{X}_{\bar{i}}, \alpha), \quad \text{where} \tag{9}$$

$$f(\boldsymbol{z}_i; \boldsymbol{X}_{\bar{i}}, \alpha) \triangleq \inf_{\boldsymbol{\gamma}_i \succcurlyeq \boldsymbol{0}} \sum_j \frac{z_{ij}^2}{\gamma_{ij}} + \log |\boldsymbol{X}_{\bar{i}} \boldsymbol{\Gamma}_i \boldsymbol{X}_{\bar{i}}^\top + \alpha \boldsymbol{I}| \tag{10}$$

is a data-dependent penalty function, parameterized by $\boldsymbol{X}_{\bar{i}}$ and α, that is only expressible in variational form.[3] Of course once we view this Bayesian model in terms of (9), we need no longer enforce that the α embedded in f equal the α scale factor found in the ℓ_2-norm error term. This emergent flexibility facilitates later analysis of f restricted to the feasible region but with arbitrary α, and contributes to a broader class of viable algorithms. Finally, although there is generally no closed-form solution for f, it nonetheless can be shown to be a strictly concave, non-decreasing function of each coefficient magnitude $|z_{ij}|$ for all $\alpha \geq 0$, and hence it naturally favors exactly sparse solutions (Wipf et al., 2011), meaning many $z_{ij} = 0$. Therefore we propose to attack (2) by replacing the ℓ_0 norm with f for each point, rather than the standard convex ℓ_1 norm alternative.

Once we have some $\hat{\boldsymbol{Z}}$ obtained by optimizing this revised penalty across all i, we may then form an affinity matrix $\boldsymbol{A}$ just as with previous methods to facilitate the subsequent spectral clustering step. We will henceforth refer to this procedure as DD-SSC, for *data-dependent sparse subspace clustering*. Like the original ℓ_1-SSC, DD-SSC naturally handles extensions to account for both outlier removal and affine subspaces.

Outlier Removal: In many applications it is quite common to have outlying data points, meaning that some columns of $\boldsymbol{X}$ do not adhere to the union of subspaces model. If such outliers are not properly accounted for, virtually all clustering algorithms, sparsity-based or not, will fail. Fortunately, the SSC framework provides a natural mechanism for removing such divisive points (Soltanolkotabi and Candes, 2012), and DD-SSC can seamlessly piggyback this agency. DD-SSC can also handle element-wise corruptions in a principled fashion, but we postpone this analysis to the supplementary file.

The basic idea proceeds as follows. First we run DD-SSC on each point $\boldsymbol{x}_i$ and compute the corresponding $\hat{\boldsymbol{z}}_i$, which we expect to be sparse. However, because the outliers do not lie in one of the low-dimensional subspaces, we envisage that the support set will be considerably larger (less sparse) than for inlier points. Hence we declare $\boldsymbol{x}_i$ to be an outlier if and only if the number of nonzero entries of $\hat{\boldsymbol{z}}_i$ surpasses a certain threshold, i.e., $\|\hat{\boldsymbol{z}}_i\|_0 \geq \tau$. Moreover, it is our hope that, presuming the DD-SSC penalty f is able to produce even greater sparsity for inlier points, the gap between normal and outlying points will be more pronounced. In Section 4, we will empirically show that this is indeed the case. Finally, once outliers have been identified, the affinity matrix is constructed using the remaining inliers for subsequent spectral clustering.

Affine Subspaces: Real-World data often lie in a union of affine subspaces, a more general model which includes linear subspaces as a special case.[4] For example, motion segmentation problems require the clustering of data that lie in the union of 3-dimensional affine subspaces (Tomasi and Kanade, 1992). To enforce an affine subspace model, it is sufficient in principle to include the additional constraint $\sum_j z_{ij} = 1$, which leads to translation invariance of SSC algorithms (Rao et al., 2010; Elhamifar and Vidal, 2013), at least assuming columns of $\boldsymbol{X}$ are not normalized (more on this in Section 3). This constraint can be incorporated into DD-SSC by updating the likelihood model (4) and then performing the associated marginalization. The net result is that we only need to replace $\boldsymbol{X}_{\bar{i}}$ and $\boldsymbol{I}$ in the objective with $\boldsymbol{X}^+$ and $\boldsymbol{I}^+$ respectively, where $\boldsymbol{X}^+ \triangleq [\boldsymbol{X}; \boldsymbol{1}_{(n)}^\top]$ (i.e., a length

[3] If some $z_{ij} = 0$, then we allow by definition the corresponding variational parameter γ_{ij} to exactly equal zero as well; given that the log-det term is a concave, non-decreasing function of each γ_{ij}, zero will in fact be the minimizing value in such a case.

[4] An affine subspace is merely defined as a standard linear subspace that has been translated away from the origin via an arbitrary offset.

n row of ones is appended to the bottom of $\boldsymbol{X}$), and $\boldsymbol{I}^+ \triangleq \left[[\boldsymbol{I}, \boldsymbol{0}_{(n-1)}]; \boldsymbol{0}_{(n)}^\top\right]$.

3 ANALYSIS OF DD-SSC

Motivating data-dependent terms for SSC: When we apply sparse regularization such as the ℓ_1 norm penalty (or essentially any other sparsity penalty in the literature), results are highly sensitive to correlation structure in the data matrix $\boldsymbol{X}$, meaning strong off-diagonal elements in $\boldsymbol{X}^\top\boldsymbol{X}$ (see (Candes et al., 2006)). In tackling this problem, we observed that it makes sense to include a data-dependent penalty that effectively compensates for this correlation structure. To see this, a simple analogy is in order. Suppose we would like to solve a regularized regression problem of the form

$$\min_{\boldsymbol{z}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{z}\|^2 + \textstyle\sum_i g(z_i), \tag{11}$$

where g is some regularizer that favors small magnitudes. Generally speaking, the solution will be highly dependent on the column norms of $\boldsymbol{X}$, meaning that if the norm of some column $\boldsymbol{x}_i$ is very small, then the corresponding coefficient z_i needs to be made large to compensate, and this will be more heavily penalized by g. Of course in this case there is a trivial solution: If we replace $g(z_i)$ with $g(\|\boldsymbol{x}_i\|z_i)$, then elements of $\boldsymbol{z}$ associated with small column norms will be penalized proportionally less, and the aggregate behavior will be exactly as though the design matrix $\boldsymbol{X}$ had normalized columns.

In this example, we have effectively used a *column-norm-dependent penalty* $\sum_i g(\|\boldsymbol{x}_i\|z_i)$ to counterbalance potentially disparate scale factors. Drawing from this experience though, if our concern is also strong off-diagonal factors in $\boldsymbol{X}^\top\boldsymbol{X}$, and not just disparate scales along the diagonal as represented by $\|\boldsymbol{x}_i\|$, it makes sense to consider some new penalty $f(\boldsymbol{z}; \boldsymbol{X}^\top\boldsymbol{X})$ that explicitly depends on this correlation structure. The subspace clustering penalty we proposed does exactly this. In particular, given the general determinant identity

$$\log\left|\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^\top + \alpha\boldsymbol{I}\right| = \log\left|\tfrac{1}{\alpha}\boldsymbol{X}^\top\boldsymbol{X} + \boldsymbol{D}^{-1}\right| + C \tag{12}$$

where $C{=}\log|\boldsymbol{D}|{+}\log|\alpha\boldsymbol{I}|$ and $\boldsymbol{D}$ is a non-negative diagonal matrix, when we optimize (10), the resulting penalty will explicitly depend on $\boldsymbol{X}^\top\boldsymbol{X}$ through the action of this volumetric log-det measure that is highly sensitive to correlations. Moreover, although there is no closed-form solution for the actual penalty, we can nonetheless analyze its behavior in the specific context of SSC, revealing both theoretically and empirically in the coming sections, that it directly solves many previously-unaddressed limitations (and the computational complexity is proportional to the original ℓ_1-SSC).

Improvement over ℓ_1-SSC: For purposes of the simplest practical deployment in a few lines of code, DD-SSC can be implemented using a form of the EM algorithm treating each latent $\boldsymbol{z}_i$ as hidden data (Dempster et al., 1981). However, an alternative implementation can be designed using iterative reweighted ℓ_1-norm updates (Wipf et al., 2011) more amenable to analysis and the elucidation of an explicit advantage of the DD-SSC penalty function over the convex ℓ_1-SSC standard, especially when correlated subspaces confound the latter.

In the present context, iterative reweighted ℓ_1 minimization, which is a specific variant of a majorization-minimization algorithm (Hunter and Lange, 2004), proceeds by forming a convex, first-order Taylor-series approximation to the non-convex penalty function f in (10). More concretely, for the i-th data point, we begin from an initial weight vector $\boldsymbol{w}^{(0)} = \boldsymbol{1}$ and then proceed to the $(t+1)$-th iteration by computing

$$\boldsymbol{z}_i^{(t+1)} \leftarrow \arg\min_{\boldsymbol{z}_i} \sum_j w_j^{(t)}|z_{ij}| \quad \text{s.t. } \boldsymbol{x}_i = \boldsymbol{X}_{\bar{i}}\boldsymbol{z}_i,$$
$$\boldsymbol{w}^{(t+1)} \leftarrow \left.\frac{\partial f(\boldsymbol{z}_i; \boldsymbol{X}_{\bar{i}}, \alpha)}{\partial|\boldsymbol{z}|}\right|_{\boldsymbol{z}=\boldsymbol{z}_i^{(t+1)}}, \tag{13}$$

where the $|\cdot|$ operator is understood to apply element-wise. Furthermore, in certain cases the required gradient is available in closed-form even though the original penalty f is not. For example, let

$$\eta_j(\boldsymbol{z}_i; \alpha, q) \triangleq \left[\boldsymbol{x}_j^\top\left(\alpha I + \boldsymbol{X}_{\bar{i}}\left|Z_i^{(t+1)}\right|^2\boldsymbol{X}_{\bar{i}}^\top\right)^{-1}\boldsymbol{x}_j\right]^q, \tag{14}$$

where $\left|Z_i^{(t+1)}\right|$ denotes a diagonal matrix with j-th diagonal entry given by $|z_{ij}^{(t+1)}|$. Then when $\alpha \to 0$, $\boldsymbol{w}^{(t+1)}$ can be updated using $w_j^{(t+1)} = \eta_j(\boldsymbol{z}_i; 0, 1/2), \quad \forall j$. Using straightforward application of results from (Sriperumbudur and Lanckriet, 2012), the resulting iterations are guaranteed to converge to the set of local minima (or possibly saddle points) of f in the feasible region $\boldsymbol{x}_i = \boldsymbol{X}_{\bar{i}}\boldsymbol{z}_i$. Additionally, for other values of α and q we obtain a valid, generalized weighting function, although it need not directly correspond with the original Bayesian model from which DD-SSC was derived.

Proceeding further, it is thus far unclear what intrinsic advantage this overall class of estimators has over the canonical ℓ_1-SSC, nor if the required $\eta_j(\boldsymbol{z}_i; 0, 1/2)$ terms are even a suitable choice of weighting factor. However, we will now demonstrate that the above generalized, iterative reweighted ℓ_1 form of DD-SSC is guaranteed to do as well or better than ℓ_1-SSC in certain circumstances.

For this purpose, we first say that a candidate solution $\hat{\boldsymbol{z}}_i$

is *subspace optimal* if its nonzero elements occur only at locations associated with data points from the subspace to which $\boldsymbol{x}_i$ belongs. Note that if an aggregate solution $\hat{\boldsymbol{Z}}$ is subspace optimal for all i, it is a trivial matter to design a clustering wrapper (spectral or otherwise), such that the correct final clustering is obtained. So clearly if we can guarantee subspace optimal sparse representations, the brunt of our work is complete.

Theorem 1 *The DD-SSC updates produced by (13) satisfy the following:*

1. *If at any iteration we compute a subspace optimal solution $\boldsymbol{z}_i^{(t)}$, then all subsequent iterations are guaranteed to be subspace optimal for any $\alpha \in (0, \alpha']$, provided α' is sufficiently small.*

2. *For any identifiable configuration of subspaces, some $q \geq 1/2$, $\alpha \in (0, \alpha']$, and α' sufficiently small, there will always exist configurations of points within each subspace such that the iterations are guaranteed to produce a subspace optimal solution for all i.*

Given that we initialize DD-SSC using $\boldsymbol{w}^{(0)} = \mathbf{1}$, the first iteration is nothing more than ℓ_1-SSC. Combined with the first property from Theorem 1, this ensures that we automatically inherit whatever desirable theoretical properties of the ℓ_1 solution that exist (Soltanolkotabi and Candes, 2012), and we need not worry about subsequent iterations diverging from a subspace correct solution found by ℓ_1-SSC. Extra iterations can only improve our chances, they can never make things worse.[5]

Moreover, the second property from above guarantees that no matter how entwined the various subspaces are, for at least some constellations of the points additional iterations will indeed improve the situation. While the proof construction contains further details (see supplementary), in brief, sufficiently clustered point clouds (lying within a region of non-zero Lebesgue measure inside each subspace), will ensure optimal recovery. In contrast, no other algorithm we are aware of satisfies a similar result. We also emphasize that the ℓ_1 norm solution itself can be quite difficult to improve upon, largely because in failure cases it is frequently positioned (often provably so) in a poor basin of attraction with respect to more complex, non-convex regularizers, *especially* when correlated data $\boldsymbol{X}$ is involved. Therefore further iterations cannot escape, unlike the DD-SSC updates.

[5]Note that prior conditions have been established based on the distribution of nonzero elements in a sparse representation such that guaranteed improvement is possible using iterative reweighted ℓ_1 minimization (Wipf and Nagarajan, 2010). However, these results are inapplicable in the present subspace clustering context and Theorem 1 above relies on an entirely different proof construction (see supplementary file).

When combined together then, we believe that these components of Theorem 1 suggest that there is minimal risk in deploying DD-SSC, but potentially much to gain, especially in challenging recovery conditions with dense, correlated subspaces that can cause existing algorithms to fail. While perhaps not immediately obvious, all of this is possible because the DD-SSC penalty function f depends on the design matrix $\boldsymbol{X}_{\bar{i}}$ in such a way that compensation for subspace structure is possible. In contrast, virtually all existing subspace clustering algorithms rely on *data-independent* penalties, e.g., the ℓ_1, ℓ_2, or nuclear norm, or the hybrid method from (Yang et al., 2016). However, one noteworthy exception is the correlation adaptive subspace segmentation algorithm (CASS) (Lu et al., 2013), which adopts the Trace Lasso penalty function $g(\boldsymbol{z}_i; \boldsymbol{X}_{\bar{i}}) = \|\boldsymbol{X}_{\bar{i}} |Z_i|\|_*$ (Grave et al., 2011). Although this selection does compensate to some extent for correlation structure to increase the number of nonzeros in the correct subspace, it is *not* a concave function of the coefficients $|z_{ij}|$ (unlike both ℓ_1-SSC and DD-SSC), and therefore subspace optimal solutions are essentially impossible to guarantee, except in the trivial case where the subspaces are independent and virtually any algorithm is subspace optimal. Additionally, CASS and other state-of-the-art algorithms do not benefit from the invariance properties discussed next.

Desirable Invariances: The original ℓ_1-SSC algorithm was motivated, at least initially, by its theoretical ability to recover maximally sparse solutions in the context of compressive sensing (CS) (Candes et al., 2006). But there remain several crucial differences between CS and SSC applications. Most importantly, CS typically relies on randomized dictionaries with suitable concentration properties in high dimensions such that different columns are roughly orthogonal with equivalent column ℓ_2 norms. But in SSC we are not free to choose the dictionaries $\boldsymbol{X}_{\bar{i}}$, rather, they are determined by the subspace structure of the data and may display both high degrees of correlation and columns with vastly different norms, both of which can severely bias ℓ_1 regularization dramatically. With regard to the latter, we could of course consider first normalizing each point $\boldsymbol{x}_i$ of the full data $\boldsymbol{X}$ by dividing with $\|\boldsymbol{x}_i\|_2$, but unlike in the CS domain, this may actually hurt performance more than it helps depending on how the data are structured.

For example, consider data generated as $\boldsymbol{X} = \boldsymbol{\Phi}\boldsymbol{S} + \beta\boldsymbol{L}$, where $\boldsymbol{\Phi}$ reflects some arbitrary subspace structure involving embedded points with unit ℓ_2 norm, $\boldsymbol{S}$ is a diagonal scaling matrix, $\boldsymbol{L}$ is a low-rank additive component, and β is a positive scalar. Now suppose we try to recover $\boldsymbol{x}_i$ using $\boldsymbol{X}_{\bar{i}}$ as the dictionary using ℓ_1-SSC. Without column normalization, the scaling via $\boldsymbol{S}$ can completely disrupt the effectiveness of the ℓ_1 norm solution

by favoring points in the wrong subspace that happen to have large magnitudes. However, if we do apply normalization and β is large, then the resulting ℓ_2 column norms will be overshadowed by the low-rank term contributing to arbitrarily bad solutions as well. Moreover, this is not some contrived situation. In the case of affine subspaces, such an additive low-rank term will always be present to enforce each subspaces' translation away from zero, namely, $\boldsymbol{L} = [\boldsymbol{b}_1\mathbf{1}_{n_1}^\top, \boldsymbol{b}_2\mathbf{1}_{n_2}^\top, \ldots]$, where each $\boldsymbol{b}_k$ is a bias vector and $\text{rank}[\boldsymbol{L}] = m$, the number of subspaces.

Further compounding the problem is that there is ambiguity in the constraint set from (2) in that $\boldsymbol{W}\boldsymbol{X} = \boldsymbol{W}\boldsymbol{X}\boldsymbol{Z}$ represents an equivalent feasible region whenever $\boldsymbol{W}$ is an invertible feature transformation. While such a $\boldsymbol{W}$ has no effect on the constraint per se, it will have a major impact on the column sizes should we choose to normalize. Hence even beyond the intrinsic subspace structure and correlation discussed above, issues of feature representations, affine translations, and column scaling can act as a major disruptive force to ℓ_1-SSC, as well as virtually all other existing subspace clustering algorithms we are aware of. For example, the CASS algorithm will be particularly sensitive to these effects since the data-dependent Trace Lasso penalty function is highly dependent on both transformations via $\boldsymbol{W}$ or translations via $\boldsymbol{L}$. Likewise for algorithms built upon IHT (Yang et al., 2016).

It is here that DD-SSC maintains another considerable advantage over existing methods as follows:

Theorem 2 *Let $\widetilde{\boldsymbol{X}} \triangleq \boldsymbol{W}\boldsymbol{X}\boldsymbol{S}$ denote a transformed and rescaled version of $\boldsymbol{X}$, where $\boldsymbol{W}$ is an arbitrary matrix and $\boldsymbol{S}$ is a diagonal matrix. Then the support set (and therefore subspace optimality) of any local or global minimizer of*

$$\min_{\boldsymbol{z}_i} f(\boldsymbol{z}_i; \widetilde{\boldsymbol{X}}_{\bar{i}}, 0) \quad \textit{s.t. } \widetilde{\boldsymbol{x}}_i = \widetilde{\boldsymbol{X}}_{\bar{i}}\boldsymbol{z}_i \tag{15}$$

is invariant to $\boldsymbol{W}$ and $\boldsymbol{S}$ provided that both are invertible.

This result follows by extending (Wipf, 2011, Lemma 1); details are omitted here. The consequences of the invariance afforded by Theorem 2 in the context of SSC are profound. First, we need not worry about feature transformations/representations and their effect on column norm scalings given that DD-SSC is *jointly* invariant to either. And secondly, assuming $\boldsymbol{L}$ is sufficiently low rank, then the projection operator onto $\text{null}[\boldsymbol{L}^\top]$, which is the orthogonal complement of $\text{range}[\boldsymbol{L}]$, will be nearly invertible. Therefore if we assign $\boldsymbol{W}$ to be an invertible approximation to this operator, we can transform $\boldsymbol{X} = \boldsymbol{\Phi}\boldsymbol{S} + \beta\boldsymbol{L}$ via

$$\boldsymbol{W}\boldsymbol{X} = \boldsymbol{W}\boldsymbol{\Phi}\boldsymbol{S} + \beta\boldsymbol{W}\boldsymbol{L} \approx \boldsymbol{W}\boldsymbol{\Phi}\boldsymbol{S}, \tag{16}$$

and so DD-SSC should be nearly invariant to affine transformations as well. Experimental results will confirm this conclusion revealing the sensitivity of ℓ_1-SSC.

4 EXPERIMENTS

Challenging Subspace Detection: In (Soltanolkotabi and Candes, 2012) a series of synthetic experiments were designed to challenge the performance of ℓ_1-SSC. We embed DD-SSC into publicly available code[6] and conduct the same experiments. We begin by testing the ability to cluster subspaces with intersection. Specifically, two subspaces of dimension $d_1 = d_2 = 10$ are embedded in $\mathbb{R}^{20}$ with an intersection of dimension $t \leq d_1$, all generated uniformly at random. Then, $n_1 = n_2 = 20d_1$ points are selected from each subspace also uniformly at random. All data are normalized to have unit ℓ_2 norm.

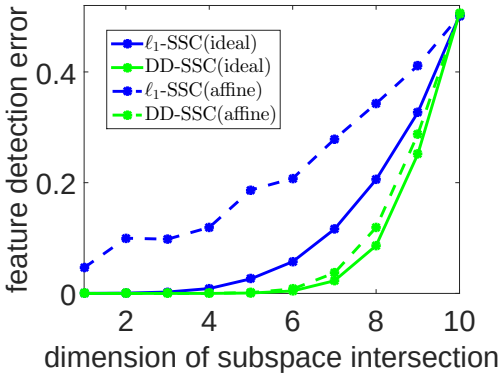


Figure 1: Feature detection error as a function of the subspace intersection dimension. Any algorithm will not have error below 0.5 on average once $t = 10$ where the two subspaces merged to one.

Although this experiment introduces some confounding structure by virtue of the subspace intersections, the remaining conditions of this experiment are rather ideal for ℓ_1-SSC since all points are effectively sampled uniformly on the unit ℓ_2 norm ball. To investigate further disruption of such idyllic conditions, we first remove the original column normalization from above and then shift each subspace away from the origin to form affine subspaces. Ultimately this means that the data matrix $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2]$ is modified to $\boldsymbol{X}\boldsymbol{S} + [\boldsymbol{b}_1\mathbf{1}_{n_1}^\top \; \boldsymbol{b}_2\mathbf{1}_{n_2}^\top]$, where $\boldsymbol{S}$ is a diagonal matrix of column scalings (which range roughly between 3 and 6 with high probability) and $\boldsymbol{b}_i \triangleq b_i\mathbf{1}_d$ with b_i sampled from a normal distribution with standard deviation 10.

In Figure 1, we show results under both the original (ideal) setting and this more challenging affine setting. Following (Soltanolkotabi and Candes, 2012), we define the *feature detection error* as $\frac{1}{n}\sum_{i=1}^{n} 1 - \frac{\|\boldsymbol{z}_{ik_i}\|_1}{\|\boldsymbol{z}_i\|_1}$,

[6] http://www-bcf.usc.edu/∼soltanol/code.html

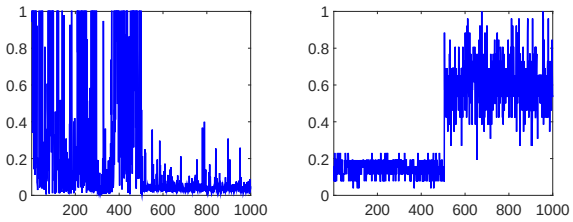
where $\boldsymbol{z}_{ik_i}$ is the portion of $\boldsymbol{z}_i$ corresponding to points from the i-th subspace and use it for evaluation. This metric goes to zero when each data point is reconstructed using only points from its own subspace, while it tends to one when each point is reconstructed using points from the other subspaces. Figure 1 displays the feature detection error averaged across 20 randomized trials.

As in (Soltanolkotabi and Candes, 2012), under the original ideal setting, we observed failure (error $\neq 0$) of ℓ_1-SSC at $t = 4$. Meanwhile DD-SSC only starts to fail at $t = 6$ and consistently maintains a lower error up to the non-identifiable limit when $t = 10$ and the two subspaces merge to one. (Note that, any possible algorithm will not have error below 0.5 on average once $t = 10$.) Moreover, the affine transformation significantly biases the estimation results of ℓ_1-SSC. ℓ_1-SSC begins to fail even with an intersection dimension of only $t = 1$, and the error continues to grow much more sharply than before (and column-normalization cannot fix the problem because of the affine component). Meanwhile DD-SSC results are essentially unchanged as predicted by our invariance theory.[7]

Outlier Detection with Synthetic Data: We now compare the performance of DD-SSC with ℓ_1-SSC in detecting outliers, noting that other algorithms producing diffuse representations such as CASS or LSR do not naturally facilitate this added flexibility, and existing theory is limited to sparsity-based approaches. We first consider the most demanding problem in (Soltanolkotabi and Candes, 2012), where $m = 20$ subspaces of dimension $d_k = 5, \forall k$ are generated in $\mathbb{R}^{50}$ uniformly at random. From each subspace, 25 points are drawn uniformly at random so that the total number of data points is 500. Data points are *not* normalized such that modest differences remain in the corresponding column norms. We then shift points away from the origin as above.

Next $n_0 = 500$ outliers, equal to the number of inlier points, are chosen uniformly at random and appended to the inliers $\boldsymbol{X}$. We then run ℓ_1-SSC and DD-SSC and compare the degree of sparsity in each recovered $\hat{\boldsymbol{z}}_i$, with the hope that inlier samples with have a lower value than the outliers as described in Section 2. For DD-SSC we simply measure sparsity via $\|\hat{\boldsymbol{z}}_i\|_0$, but we observed empirically that ℓ_1-SSC works poorly with this metric. So instead, consistent with (Soltanolkotabi and Candes, 2012) we report $\|\hat{\boldsymbol{z}}_i\|_1$ values for ℓ_1-SSC.

Figure 2 displays the results. All plots were rescaled for visualization purposes. Clearly DD-SSC's invariance to scalings and translations is also a significant advantage when it comes to locating outliers, with a clear distinction between inliers (points on the left half of each subplot) and outliers (points on the right half), and a simple threshold around 0.3 would nearly resolve every instance. In contrast, ℓ_1-SSC almost completely fails for any threshold.



(a) ℓ_1-SSC (b) DD-SSC

Figure 2: Outlier detection in affine subspaces via sparsity measurements.

Clustering Accuracy with Real Data: Following the literature, we test DD-SSC on two challenging benchmarks, the Extended Yale B (Wright et al., 2009) and the MNIST,[8] using the same testing protocol from (Lu et al., 2013). The Yale B consists of 2,432 frontal face images of 38 subjects under various lighting, poses and illumination conditions. Here we use the first 10 subjects' face images for subspace clustering (which have been previously shown to be most difficult), after first projecting onto a 60-dimensional subspace using PCA. The MNIST benchmark includes the handwritten digits 0-9 from 10 subjects. Following (Lu et al., 2013), we select a subset consisting of the first 50 samples from each subject.

Table 1: Final clustering accuracies (%) on the Extended Yale B and the MNIST data sets.

	kNN	LRR	LSR	CASS	ℓ_1-SSC	DD-SSC
YaleB	50.94	65.00	73.59	81.88	78.44	**84.84**
MNIST	61.00	66.80	68.00	73.80	71.60	**75.40**

In Table 1, we report clustering accuracies of DD-SSC, after a final spectral clustering step for both benchmarks. Reported performances from (Lu et al., 2013) for other algorithms under equivalent conditions are included for comparison purposes. These include low rank representation (LRR) (Liu et al., 2013), least-squares representation (LSR) (Lu et al., 2012), and CASS (Lu et al., 2013), and a *k*-Nearest Neighbor (kNN) baseline. Because outliers are absent, non-adversarial feature representations are used, and the final clustering accuracies are largely influenced by external factors and post-processing, these data sets do not fully showcase the ability of DD-SSC. Regardless, our method displays the best performance.

[7] As a side note, IHT iterations applied to an ℓ_1-SSC initialization, such as proposed in (Yang et al., 2016), cannot appreciably improve results over ℓ_1-SSC alone in these experiments (not shown).

[8] http://yann.lecun.com/exdb/mnist/

Outlier Detection in Motion Segmentation Data: A principled mechanism for generating outlying trajectories for the Hopkins 155 data[9] has been introduced in (Rao et al., 2010). Specifically, outliers were produced by choosing a random initial point in the first frame and then selecting a random increment between successive frames. Each increment is generated by taking the difference between the coordinates of a randomly chosen point in two randomly chosen consecutive frames. In this way the outlying trajectories may qualitatively have the same statistical properties as the other trajectories, but will not be consistent with any particular motion model. Here we examine the most difficult sequence "1R2RC" for evaluation with outliers.

In Table 2, we show the performance of ℓ_1-SSC and DD-SSC using the outlier detection strategy described in Section 2. Given a varying number n_0 of outliers have been added, we sort the learned representations for all points via the sparsity measures, i.e. ℓ_1 norm for ℓ_1-SSC and ℓ_0 norm for DD-SSC; the highest n_0 are then declared outliers and compared to ground truth. To evaluate the performance, we define a detection accuracy as $\frac{\#\ correctly\ found\ inliers}{\#\ total\ inliers}$. When we increase the number of outliers (measured by its percentage with regard to the number of inliers), ℓ_1-SSC gradually failed to find all correct inliers. Meanwhile, DD-SSC can achieve almost 100% success even when there are twice as many outliers as the number of inliers. Typical ROC curves can be found in the supplementary file.

Table 2: Outlier detection on Hopkins 155 motion data.

[%]	0	15	30	50	100	150	200
ℓ_1-SSC	1.00	0.91	0.83	0.77	0.61	0.49	0.43
DD-SSC	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.99**

Comparisons with (Yang et al., 2016) Using Real Data: As described in Section 1, the very recent approach from (Yang et al., 2016) attempts to approximately solve (2) by first computing an ℓ_1 solution and then later refining it via IHT iterations. Note that IHT is just a projected gradient method for locally minimizing the ℓ_0 norm, and the particular extrema to which it converges will be highly sensitive to the types of correlation structure we described in Section 3 (Blumensath and Davies, 2009). Moreover, in many circumstances it can be proven that the ℓ_1 norm initialization will be at or near a local minima of the IHT objective, in which case improvement is not even feasible. Still (Yang et al., 2016) nonetheless presents some nice theory for when this proposed *approximate* ℓ_0-SSC pipeline, termed Aℓ_0-SSC, is likely to produce good solutions; however, the required, idealized conditions on the data $\boldsymbol{X}$ are similar to those in the compressive sensing literature and disallow the types of correlations commonly found in clustering problems.

[9] http://www.vision.jhu.edu/data/hopkins155/

Table 3: Final clustering accuracies (%) compared with Aℓ_0-SSC on UCI and COIL data sets.

	SM CE	OMP-SSC	ℓ_1-SSC	Aℓ_0-SSC	DD-SSC
Ionosphere	68.09	63.53	51.28	76.92	**84.90**
Heart	59.63	55.19	63.70	64.44	**77.41**
COIL-20	75.49	33.89	78.54	84.72	**90.00**
COIL-100	56.39	16.67	52.75	76.83	**80.83**

We compare the capability of both Aℓ_0-SSC and DD-SSC to improve upon ℓ_1-SSC using representative experiments from (Yang et al., 2016) involving real-world UCI and COIL data sets. Specifically, the UCI Ionosphere data contains 351 data points from 2 classes of dimensionality 34, while UCI Heart data contains 270 points from 2 classes of dimensionality 13. The COIL-20 and COIL-100 databases have respectively 20 and 100 objects with 72 images of size 32×32 for each object, and therefore 1440 and 7200 total images overall. The images were taken 5 degrees apart as an object was rotated on a turntable.

Our comparisons with Aℓ_0-SSC are shown is Table 3, together with the results of two additional competing algorithms included in (Yang et al., 2016), namely, sparse manifold clustering and embedding (SMCE) Elhamifar and Vidal (2011), and orthogonal matching pursuit or OMP-SSC Dyer et al. (2013), which applies a greedy sparse estimation strategy. Note that, for COIL-20 and COIL-100, we provide results using all 20 and 100 clusters respectively, the hardest cases, and along with the Ionosphere and Heart data, these arguably represent the most demanding experimental conditions from (Yang et al., 2016). As shown in Table 3, our model outperforms Aℓ_0-SSC. Moreover, we observe that in the most difficult case, the Heart data, Aℓ_0-SSC is not able to significantly improve upon the ℓ_1-SSC solution, unlike DD-SSC which consistently supplies an advantage.

5 CONCLUSION

Sparsity promoting algorithms such as what we have advocated are certainly not new. However, deployment of our particular proposal in the context of subspace clustering is firmly supported by the novel theoretical arguments and strong, state-of-the-art empirical evidence presented herein. Simply put, the DD-SSC pipeline displays a remarkable degree of invariance to the very types of confounding factors, e.g., dictionary structures, distracting feature transformations, translations, and outlying data points etc., that otherwise derail existing segmentation algorithms.

References

S. Babacan, S. Nakajima, and M. Do. Probabilistic low-rank subspace clustering. In *Advances in Neural Information Processing Systems*, pages 2744–2752, 2012.

T. Blumensath and M.E. Davies. Iterative thresholding for sparse approximations. *J. Fourier Analysis and Applications*, 14(5), 2008.

T. Blumensath and M.E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3), 2009.

E. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489–509, 2006.

A. Dempster, D. Rubin, and R. Tsutakawa. Estimation in covariance components models. *Journal of the American Statistical Association*, 76(374):341–353, June 1981.

Eva L Dyer, Aswin C Sankaranarayanan, and Richard G Baraniuk. Greedy feature selection for subspace clustering. *Journal of Machine Learning Research*, 14(1):2487–2517, 2013.

E. Elhamifar and R. Vidal. Sparse subspace clustering. In *Computer Vision and Pattern Recognition, CVPR 2009.*, pages 2790–2797. IEEE, 2009.

E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2765–2781, 2013.

Ehsan Elhamifar and René Vidal. Sparse manifold clustering and embedding. In *Advances in neural information processing systems*, pages 55–63, 2011.

J. Feng, Z. Lin, H. Xu, and S. Yan. Robust subspace segmentation with block-diagonal prior. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3818–3825. IEEE, 2014.

E. Grave, G. Obozinski, and F. Bach. Trace lasso: a trace norm regularization for correlated designs. In *Advances in Neural Information Processing Systems*, 2011.

D. Hunter and K. Lange. A tutorial on mm algorithms. *The American Statistician*, 58(1):30–37, 2004.

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)*, pages 663–670, 2010.

G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):171–184, 2013.

C. Lu, H. Min, Z. Zhao, L. Zhu, D. Huang, and S. Yan. Robust and efficient subspace segmentation via least squares regression. In *Computer Vision–ECCV 2012*, pages 347–360. Springer, 2012.

C. Lu, J. Feng, Z. Lin, and S. Yan. Correlation adaptive subspace segmentation by trace lasso. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1345–1352. IEEE, 2013.

U. Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

D. MacKay. Bayesian interpolation. *Neural computation*, 4(3): 415–447, 1992.

S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(10):1832–1845, 2010.

M. Soltanolkotabi and E. Candes. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, 40 (4):2195–2238, 2012.

M. Soltanolkotabi, E. Elhamifar, and E. Candes. Robust subspace clustering. *The Annals of Statistics*, 42(2):669–699, 2014.

B. Sriperumbudur and G. Lanckriet. A proof of convergence of the concave-convex procedure using zangwill's theory. *Neural computation*, 24(6):1391–1407, 2012.

Michael E Tipping. Sparse bayesian learning and the relevance vector machine. *The journal of machine learning research*, 1:211–244, 2001.

C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2), 1992.

Y. Wang, D. Wipf, Q. Ling, W. Chen, and I. Wassell. Multi-task learning for subspace segmentation. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1209–1217, 2015.

D. Wipf and S. Nagarajan. Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):317–329, 2010.

D. Wipf, B. Rao, and S. Nagarajan. Latent variable bayesian models for promoting sparsity. *Information Theory, IEEE Transactions on*, 57(9):6236–6255, 2011.

D.P. Wipf. Sparse estimation with structured dictionaries. *Advances in Neural Information Processing 24*, 2011.

J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2): 210–227, 2009.

Y. Yang, J. Feng, N. Jojic, J. Yang, and T. Huang. ℓ_0-sparse subspace clustering. In *Computer Vision–ECCV 2016*. Springer, 2016.

C. You and R. Vidal. Geometric conditions for subspace-sparse recovery. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1585–1593, 2015.