Distributed Low-rank Subspace Segmentation

Ameet Talwalkar[a] Lester Mackey[b] Yadong Mu[c] Shih-Fu Chang[c] Michael I. Jordan[a]
[a]University of California, Berkeley [b]Stanford University [c]Columbia University
{ameet, jordan}@cs.berkeley.edu, lmackey@stanford.edu, {muyadong, sfchang}@ee.columbia.edu

Abstract

Vision problems ranging from image clustering to motion segmentation to semi-supervised learning can naturally be framed as subspace segmentation *problems, in which one aims to recover multiple low-dimensional subspaces from noisy and corrupted input data. Low-Rank Representation (LRR), a convex formulation of the subspace segmentation problem, is provably and empirically accurate on small problems but does not scale to the massive sizes of modern vision datasets. Moreover, past work aimed at scaling up low-rank matrix factorization is not applicable to LRR given its non-decomposable constraints. In this work, we propose a novel divide-and-conquer algorithm for large-scale subspace segmentation that can cope with LRR's non-decomposable constraints and maintains LRR's strong recovery guarantees. This has immediate implications for the scalability of subspace segmentation, which we demonstrate on a benchmark face recognition dataset and in simulations. We then introduce novel applications of LRR-based subspace segmentation to large-scale semi-supervised learning for multimedia event detection, concept detection, and image tagging. In each case, we obtain state-of-the-art results and order-of-magnitude speed ups.*

1. Introduction

Visual data, though innately high dimensional, often reside in or lie close to a union of low-dimensional subspaces. These subspaces might reflect physical constraints on the objects comprising images and video (e.g., faces under varying illumination [2] or trajectories of rigid objects [24]) or naturally occurring variations in production (e.g., digits hand-written by different individuals [12]). *Subspace segmentation* techniques model these classes of data by recovering bases for the multiple underlying subspaces [10, 7]. Applications include image clustering [7], segmentation of images, video, and motion [30, 6, 26], and affinity graph construction for semi-supervised learning [32].

One promising, convex formulation of the subspace segmentation problem is the *low-rank representation* (LRR) program of Liu et al. [17, 18]:

$$(\hat{\mathbf{Z}}, \hat{\mathbf{S}}) = \operatorname*{argmin}_{\mathbf{Z},\mathbf{S}} \quad \|\mathbf{Z}\|_* + \lambda\|\mathbf{S}\|_{2,1} \qquad (1)$$
$$\text{subject to} \quad \mathbf{M} = \mathbf{M}\mathbf{Z} + \mathbf{S}\,.$$

Here, $\mathbf{M}$ is an input matrix of datapoints drawn from multiple subspaces, $\|\cdot\|_*$ is the nuclear norm, $\|\cdot\|_{2,1}$ is the sum of the column ℓ_2 norms, and λ is a parameter that trades off between these penalties. LRR segments the columns of $\mathbf{M}$ into subspaces using the solution $\hat{\mathbf{Z}}$, and, along with its extensions (e.g., LatLRR [19] and NNLRS [32]), admits strong guarantees of correctness and strong empirical performance in clustering and graph construction applications. However, the standard algorithms for solving Eq. (1) are unsuitable for large-scale problems, due to their sequential nature and their reliance on the repeated computation of costly truncated SVDs.

Much of the computational burden in solving LRR stems from the nuclear norm penalty, which is known to encourage low-rank solutions, so one might hope to leverage the large body of past work on parallel and distributed matrix factorization [11, 23, 8, 31, 21] to improve the scalability of LRR. Unfortunately, these techniques are tailored to optimization problems with losses and constraints that decouple across the entries of the input matrix. This decoupling requirement is violated in the LRR problem due to the $\mathbf{M} = \mathbf{M}\mathbf{Z} + \mathbf{S}$ constraint of Eq. (1), and this non-decomposable constraint introduces new algorithmic and analytic challenges that do not arise in decomposable matrix factorization problems.

To address these challenges, we develop, analyze, and evaluate a provably accurate divide-and-conquer approach to large-scale subspace segmentation that specifically accounts for the non-decomposable structure of the LRR problem. Our contributions are three-fold:

Algorithm: We introduce a parallel, divide-and-conquer approximation algorithm for LRR that is suitable for large-scale subspace segmentation problems. Scalability is achieved by dividing the original LRR problem into computationally tractable and communication-free subproblems, solving the subproblems in parallel, and combining the results using a technique from randomized matrix approximation. Our algorithm, which we call DFC-LRR, is based on the principles of the Divide-Factor-Combine (DFC) framework [21] for decomposable matrix factorization but can cope with the non-decomposable constraints of LRR.

Analysis: We characterize the segmentation behavior of our new algorithm, showing that DFC-LRR maintains the segmentation guarantees of the original LRR algorithm with high probability, even while enjoying substantial speed-ups over its namesake. Our new analysis features a significant broadening of the original LRR theory to treat the richer class of LRR-type subproblems that arise in DFC-LRR. Moreover, since our ultimate goal is subspace segmentation and not matrix recovery, our theory guarantees correctness under a more substantial reduction of problem complexity than the work of [21] (see Sec. 3.2 for more details).

Applications: We first present results on face clustering and synthetic subspace segmentation to demonstrate that DFC-LRR achieves accuracy comparable to LRR in a fraction of the time. We then propose and validate a novel application of the LRR methodology to large-scale graph-based semi-supervised learning. While LRR has been used to construct affinity graphs for semi-supervised learning in the past [4, 32], prior attempts have failed to scale to the sizes of real-world datasets. Leveraging the favorable computational properties of DFC-LRR, we propose a scalable strategy for constructing such subspace affinity graphs. We apply our methodology to a variety of computer vision tasks – multimedia event detection, concept detection, and image tagging – demonstrating an order of magnitude improvement in speed and accuracy that exceeds the state of the art.

The remainder of the paper is organized as follows. In Section 2 we first review the low-rank representation approach to subspace segmentation and then introduce our novel DFC-LRR algorithm. Next, we present our theoretical analysis of DFC-LRR in Section 3. Section 4 highlights the accuracy and efficiency of DFC-LRR on a variety of computer vision tasks. We present subspace segmentation results on simulated and real-world data in Section 4.1. In Section 4.2 we present our novel application of DFC-LRR to graph-based semi-supervised learning problems, and we conclude in Section 5.

Notation Given a matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$, we define $\mathbf{U}_M\mathbf{\Sigma}_M\mathbf{V}_M^\top$ as the compact singular value decomposition (SVD) of $\mathbf{M}$, where $\mathrm{rank}(\mathbf{M}) = r$, $\mathbf{\Sigma}_M$ is a diagonal matrix of the r non-zero singular values and $\mathbf{U}_M \in \mathbb{R}^{m\times r}$ and $\mathbf{V}_M \in \mathbb{R}^{n\times r}$ are the associated left and right singular vectors of $\mathbf{M}$. We denote the orthogonal projection onto the column space of $\mathbf{M}$ as $\mathbf{P}_M$.

2. Divide-and-Conquer Segmentation

In this section, we review the LRR approach to subspace segmentation and present our novel algorithm, DFC-LRR.

2.1. Subspace Segmentation via LRR

In the *robust subspace segmentation* problem, we observe a matrix $\mathbf{M} = \mathbf{L}_0 + \mathbf{S}_0 \in \mathbb{R}^{m\times n}$, where the columns of $\mathbf{L}_0$ are datapoints drawn from multiple independent subspaces,[1] and $\mathbf{S}_0$ is a column-sparse outlier matrix. Our goal is to identify the subspace associated with each column of $\mathbf{L}_0$, despite the potentially gross corruption introduced by $\mathbf{S}_0$. An important observation for this task is that the projection matrix $\mathbf{V}_{L_0}\mathbf{V}_{L_0}^\top$ for the row space of $\mathbf{L}_0$, sometimes termed the *shape iteration matrix*, is block diagonal whenever the columns of $\mathbf{L}_0$ lie in multiple independent subspaces [10]. Hence, we can achieve accurate segmentation by first recovering the row space of $\mathbf{L}_0$.

The LRR approach of [17] seeks to recover the row space of $\mathbf{L}_0$ by solving the convex optimization problem presented in Eq. (1). Importantly, the LRR solution comes with a guarantee of correctness: the column space of $\hat{\mathbf{Z}}$ is exactly equal to the row space of $\mathbf{L}_0$ whenever certain technical conditions are met [18] (see Sec. 3 for more details).

Moreover, as we will show in this work, LRR is also well-suited to the construction of affinity graphs for semi-supervised learning. In this setting, the goal is to define an affinity graph in which nodes correspond to data points and edge weights exist between nodes drawn from the same subspace. LRR can thus be used to recover the block-sparse structure of the graph's affinity matrix, and these affinities can be used for semi-supervised label propagation.

2.2. Divide-Factor-Combine LRR (DFC-LRR)

We now present our scalable divide-and-conquer algorithm, called DFC-LRR, for LRR-based subspace segmentation. DFC-LRR extends the principles of the DFC framework of [21] to a new non-decomposable problem. The DFC-LRR algorithm is summarized in Algorithm 1, and we next describe each step in further detail.

D step - Divide input matrix into submatrices: DFC-LRR randomly partitions the columns of $\mathbf{M}$ into t l-column submatrices, $\{\mathbf{C}_1, \ldots, \mathbf{C}_t\}$. For simplicity, we assume that t divides n evenly.

F step - Factor submatrices in parallel: DFC-LRR solves t subproblems in parallel. The ith LRR subproblem is of the form

$$\begin{aligned} \min_{\mathbf{Z}_i,\mathbf{S}_i} \quad & \|\mathbf{Z}_i\|_* + \lambda\|\mathbf{S}_i\|_{2,1} \\ \text{subject to} \quad & \mathbf{C}_i = \mathbf{M}\mathbf{Z}_i + \mathbf{S}_i\,, \end{aligned} \tag{2}$$

[1] Subspaces are *independent* if the dimension of their direct sum is the sum of their dimensions.

where the input matrix $\mathbf{M}$ is used as a dictionary but only a subset of columns is used as the observations.[2] A typical LRR algorithm can be easily modified to solve Eq. (2) and will return a low-rank estimate $\hat{\mathbf{Z}}_i$ in factored form.

C step - Combine submatrix estimates: DFC-LRR generates a final approximation $\hat{\mathbf{Z}}^{proj}$ to the low-rank LRR solution $\hat{\mathbf{Z}}$ by projecting $[\hat{\mathbf{Z}}_1, \ldots, \hat{\mathbf{Z}}_t]$ onto the column space of $\hat{\mathbf{Z}}_1$. This *column projection* technique is commonly used to produce randomized low-rank matrix factorizations [15] and was also employed by the DFC-PROJ algorithm of [21].

Runtime: As noted in [21], many state-of-the-art solvers for nuclear-norm regularized problems like Eq. (1) have $\Omega(mnk_M)$ per-iteration time complexity due to the rank-k_M truncated SVD required on each iteration. DFC-LRR reduces this per-iteration complexity significantly and requires just $\mathrm{O}(mlk_{C_i})$ time for the ith subproblem. Performing the subsequent column projection step is relatively cheap computationally, since an LRR solver can return its solution in factored form. Indeed, if we define $k' \triangleq \max_i k_{C_i}$, then the column projection step of DFC-LRR requires only $\mathrm{O}(mk'^2 + lk'^2)$ time.

Algorithm 1 DFC-LRR

```
Input: $\mathbf{M}$, $t$
$\{\mathbf{C}_i\}_{1\le i\le t}$ = SAMPLECOLS($\mathbf{M}$, $t$)
do in parallel
    $\hat{\mathbf{Z}}_1$ = LRR$(\mathbf{C}_1, \mathbf{M})$
        $\vdots$
    $\hat{\mathbf{Z}}_t$ = LRR$(\mathbf{C}_t, \mathbf{M})$
end do
$\hat{\mathbf{Z}}^{proj}$ = COLPROJ($[\hat{\mathbf{Z}}_1, \ldots, \hat{\mathbf{Z}}_t], \hat{\mathbf{Z}}_1$)
```

3. Theoretical Analysis

Despite the significant reduction in computational complexity, DFC-LRR provably maintains the strong theoretical guarantees of the LRR algorithm. To make this statement precise, we first review the technical conditions for accurate row space recovery required by LRR.

3.1. Conditions for LRR Correctness

The LRR analysis of Liu et al. [18] relies on two key quantities, the rank of the clean data matrix $\mathbf{L}_0$ and the *coherence* [22] of the singular vectors $\mathbf{V}_{L_0}$. We combine these properties into a single definition:

Definition 1 ((μ, r)-Coherence). *A matrix* $\mathbf{L} \in \mathbf{R}^{m\times n}$ *is* (μ, r)-coherent *if* $\operatorname{rank}(\mathbf{L}) = r$ *and*

$$\frac{n}{r}\|\mathbf{V}_L^\top\|_{2,\infty}^2 \le \mu,$$

where $\|\cdot\|_{2,\infty}$ *is the maximum column* ℓ_2 *norm.*[3]

Intuitively, when the coherence μ is small, information is well-distributed across the rows of a matrix, and the row space is easier to recover from outlier corruption. Using these properties, Liu et al. [18] established the following recovery guarantee for LRR.

Theorem 2 ([18]). *Suppose that* $\mathbf{M} = \mathbf{L}_0 + \mathbf{S}_0 \in \mathbb{R}^{m\times n}$ *where* $\mathbf{S}_0$ *is supported on* γn *columns,* $\mathbf{L}_0$ *is* $(\frac{\mu}{1-\gamma}, r)$*-coherent, and* $\mathbf{L}_0$ *and* $\mathbf{S}_0$ *have independent column support with* $\operatorname{range}(\mathbf{L}_0) \cap \operatorname{range}(\mathbf{S}_0) = \{\mathbf{0}\}$*. Let* $\hat{\mathbf{Z}}$ *be a solution returned by LRR. Then there exists a constant* γ^* *(depending on* μ *and* r*) for which the column space of* $\hat{\mathbf{Z}}$ *exactly equals the row space of* $\mathbf{L}_0$ *whenever* $\lambda = 3/(7\|\mathbf{M}\|\sqrt{\gamma^* l})$ *and* $\gamma \le \gamma^*$*.*

In other words, LRR can exactly recover the row space of $\mathbf{L}_0$ even when a constant fraction γ^* of the columns has been corrupted by outliers. As the rank r and coherence μ shrink, γ^* grows allowing greater outlier tolerance.

3.2. High Probability Subspace Segmentation

Our main theoretical result shows that, with high probability and under the same conditions that guarantee the accuracy of LRR, DFC-LRR also exactly recovers the row space of $\mathbf{L}_0$. Recall that in our independent subspace setting accurate row space recovery is tantamount to correct segmentation of the columns of $\mathbf{L}_0$. The proof of our result, which generalizes the LRR analysis of [18] to a broader class of optimization problems and adapts the DFC analysis of [21], can be found in the appendix.

Theorem 3. *Fix any failure probability* $\delta > 0$*. Under the conditions of Thm, 2, let* $\hat{\mathbf{Z}}^{proj}$ *be a solution returned by* DFC-*LRR. Then there exists a constant* γ^* *(depending on* μ *and* r*) for which the column space of* $\hat{\mathbf{Z}}^{proj}$ *exactly equals the row space of* $\mathbf{L}_0$ *whenever* $\lambda = 3/(7\|\mathbf{M}\|\sqrt{\gamma^* l})$ *for each* DFC-*LRR subproblem,* $\gamma \le \gamma^*$*, and* $t = n/l$ *for*

$$l \ge cr\mu\log(4n/\delta)/(\gamma^* - \gamma)^2$$

and c *a fixed constant larger than 1.*

[2] An alternative formulation involves replacing both instances of $\mathbf{M}$ with $\mathbf{C}_i$ in Eq. (1). The resulting low-rank estimate $\hat{\mathbf{Z}}_i$ would have dimensions $l \times l$, and the C step of DFC-LRR would compute a low-rank approximation on the block-diagonal matrix diag$(\hat{\mathbf{Z}}_1, \hat{\mathbf{Z}}_2, \ldots, \hat{\mathbf{Z}}_t)$.

[3] Although [18] uses the notion of column coherence to analyze LRR, we work with the closely related notion of (μ, r)-coherence for ease of notation in our proofs. Moreover, we note that if a rank-r matrix $\mathbf{L} \in \mathbb{R}^{m\times n}$ is supported on $(1-\gamma)n$ columns then the column coherence of $\mathbf{V}_L$ is μ if and only if $\mathbf{V}_L$ is $(\mu/(1-\gamma), r)$-coherent.

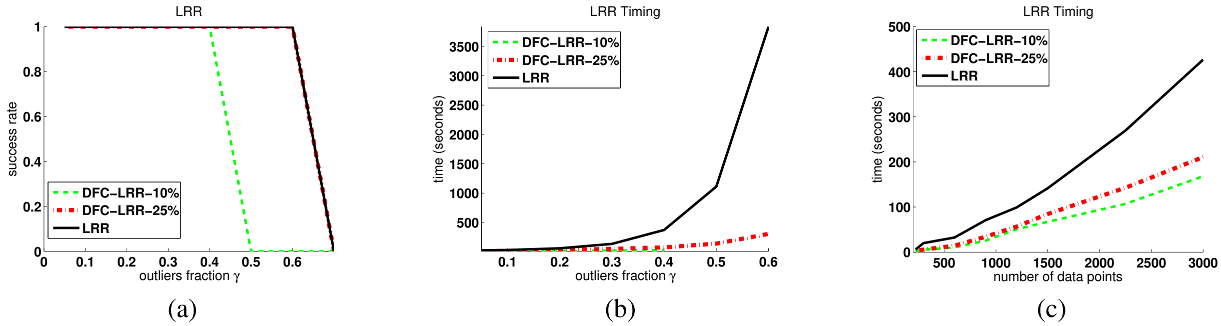


Figure 1: Results on synthetic data (reported results are averages over 10 trials). (a) Phase transition of LRR and DFC-LRR. (b,c) Timing results of LRR and DFC-LRR as functions of γ and n respectively.

Thm. 3 establishes that, like LRR, DFC-LRR can tolerate a constant fraction of its data points being corrupted and still recover the correct subspace segmentation of the clean data points with high probability. When the number of datapoints n is large, solving LRR directly may be prohibitive, but DFC-LRR need only solve a collection of small, tractable subproblems. Indeed, Thm. 3 guarantees high probability recovery for DFC-LRR even when the subproblem size l is logarithmic in n. The corresponding reduction in computational complexity allows DFC-LRR to scale to large problems with little sacrifice in accuracy.

Notably, this column sampling complexity is better than that established by [21] in the matrix factorization setting: we require $\mathrm{O}(r \log n)$ columns sampled, while [21] requires in the worst case $\Omega(n)$ columns for matrix completion and $\Omega((r \log n)^2)$ for robust matrix factorization.

4. Experiments

We now explore the empirical performance of DFC-LRR on a variety of simulated and real-world datasets, first for the traditional task of robust subspace segmentation and next for the more complex task of graph-based semi-supervised learning. Our experiments are designed to show the effectiveness of DFC-LRR both when the theory of Section 3 holds and when it is violated. Our synthetic datasets satisfy the theoretical assumptions of low rank, incoherence, and a small fraction of corrupted columns, while our real-world datasets violate these criteria.

For all of our experiments we use the inexact Augmented Lagrange Multiplier (ALM) algorithm of [17] as our base LRR algorithm. For the subspace segmentation experiments, we set the regularization parameter to the values suggested in previous works [18, 17], while in our semi-supervised learning experiments we set it to $1/\sqrt{\max{(m,n)}}$ as suggested in prior work.[4] In all experiments we report parallel running times for DFC-LRR, i.e., the time of the longest running subproblem plus the time required to combine submatrix estimates via column projection. All experiments were implemented in Matlab. The simulation studies were run on an x86-64 architecture using a single 2.60 Ghz core and 30GB of main memory, while the real data experiments were performed on an x86-64 architecture equipped with a 2.67GHz 12-core CPU and 64GB of main memory.

4.1. Subspace Segmentation: LRR vs. DFC-LRR

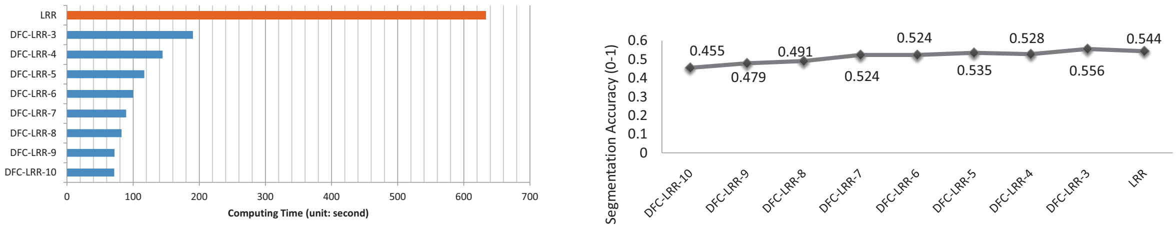
We first aim to verify that DFC-LRR produces accuracy comparable to LRR in significantly less time, both in synthetic and real-world settings. We focus on the standard robust subspace segmentation task of identifying the subspace associated with each input datapoint.

4.1.1 Simulations

To construct our synthetic robust subspace segmentation datasets, we first generate n_s datapoints from each of k independent r-dimensional subspaces of $\mathbb{R}^m$, in a manner similar to [18]. For each subspace i, we independently select a basis $\mathbf{U}_i$ uniformly from all matrices in $\mathbb{R}^{m\times r}$ with orthonormal columns and a matrix $\mathbf{T}_i \in \mathbb{R}^{r\times n_s}$ of independent entries each distributed uniformly in $[0, 1]$. We form the matrix $\mathbf{X}_i \in \mathbb{R}^{m\times n_s}$ of samples from subspace i via $\mathbf{X}_i = \mathbf{U}_i\mathbf{T}_i$ and let $\mathbf{X}_0 \in \mathbb{R}^{m\times kn_s} = [\mathbf{X}_1 \ \ldots \ \mathbf{X}_k]$. For a given outlier fraction γ we next generate an additional $n_o = \frac{\gamma}{1-\gamma}kn_s$ independent outlier samples, denoted by $\mathbf{S} \in \mathbb{R}^{m\times n_o}$. Each outlier sample has independent $\mathcal{N}(0, \sigma^2)$ entries, where σ is the average absolute value of the entries of the kn_s original samples. We create the input matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$, where $n = kn_s + n_o$, as a random permutation of the columns of $[\mathbf{X}_0 \quad \mathbf{S}]$.

In our first experiments we fix $k = 3$, $m = 1500$, $r = 5$, and $n_s = 200$, set the regularizer to $\lambda = 0.2$, and vary the fraction of outliers. We measure with what frequency LRR and DFC-LRR are able to recover of the row space of $\mathbf{X}_0$

[4] http://perception.csl.illinois.edu/matrix-rank



(a) (b)

Figure 3: Trade-off between computation and segmentation accuracy on face recognition experiments. All results are obtained by averaging across 100 independent runs. (a) Run time of LRR and DFC-LRR with varying number of subproblems. (b) Segmentation accuracy for these same experiments.

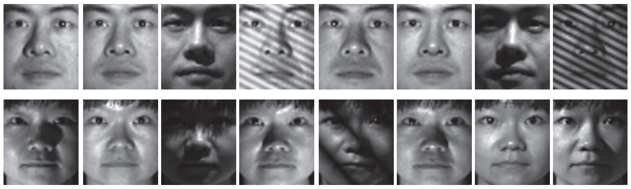


Figure 2: Exemplar face images from Extended Yale Database B. Each row shows randomly selected images for a human subject.

and identify the outlier columns in $\mathbf{S}$, using the same criterion as defined in [18].[5] Figure 1(a) shows average performance over 10 trials. We see that DFC-LRR performs quite well, as the gaps in the phase transitions between LRR and DFC-LRR are small when sampling 10% of the columns (i.e., $t = 10$) and are virtually non-existent when sampling 25% of the columns (i.e., $t = 4$).

Figure 1(b) shows corresponding timing results for the accuracy results presented in Figure 1(a). These timing results show substantial speedups in DFC-LRR relative to LRR with a modest tradeoff in accuracy as denoted in Figure 1(a). Note that we only report timing results for values of γ for which DFC-LRR was successful in all 10 trials, i.e., for which the success rate equaled 1.0 in Figure 1(a). Moreover, Figure 1(c) shows timing results using the same parameter values, except with a fixed fraction of outliers ($\gamma = 0.1$) and a variable number of samples in each subspace, i.e., n_s ranges from 75 to 1000. These timing results also show speedups with minimal loss of accuracy, as in all of these timing experiments, LRR and DFC-LRR were successful in all trials using the same criterion defined in [18] and used in our phase transition experiments of Figure 1(a).

4.1.2 Face Clustering

We next demonstrate the comparable quality and increased performance of DFC-LRR relative to LRR on real data, namely, a subset of Extended Yale Database B,[6] a standard face benchmarking dataset. Following the experimental setup in [17], 640 frontal face images of 10 human subjects are chosen, each of which is resized to be 48×42 pixels and forms a 2016-dimensional feature vector. As noted in previous work [3], a low-dimensional subspace can be effectively used to model face images from one person, and hence face clustering is a natural application of subspace segmentation. Moreover, as illustrated in Figure 2, a significant portion of the faces in this dataset are "corrupted" by shadows, and hence this collection of images is an ideal benchmark for *robust* subspace segmentation.

As in [17], we use the feature vector representation of these images to create a 2016×640 dictionary matrix, $\mathbf{M}$, and run both LRR and DFC-LRR with the parameter λ set to 0.15. Next, we use the resulting low-rank coefficient matrix $\hat{\mathbf{Z}}$ to compute an affinity matrix $\mathbf{U}_{\hat{Z}}\mathbf{U}_{\hat{Z}}^\top$, where $\mathbf{U}_{\hat{Z}}$ contains the top left singular vectors of $\hat{\mathbf{Z}}$. The affinity matrix is used to cluster the data into $k = 10$ clusters (corresponding to the 10 human subjects) via spectral embedding (to obtain a 10D feature representation) followed by k-means. Following [17], the comparison of different clustering methods relies on *segmentation accuracy*. Each of the 10 clusters is assigned a label based on majority vote of the ground truth labels of the points assigned to the cluster. We evaluate clustering performance of both LRR and DFC-LRR by computing segmentation accuracy as in [17], i.e., each cluster is assigned a label based on majority vote of the ground truth labels of the points assigned to the cluster. The segmentation accuracy is then computed by averaging the percentage of correctly classified data over all classes.

Figures 3(a) and 3(b) show the computation time and the

[5] Success is determined by whether the oracle constraints of Eq. (8) in the Appendix are satisfied within a tolerance of 10^{-4}.

[6] http://vision.ucsd.edu/~leekc/ExtYaleDatabase

segmentation accuracy, respectively, for LRR and for DFC-LRR with varying numbers of subproblems (i.e., values of t). On this relatively-small data set ($n = 640$ faces), LRR requires over 10 minutes to converge. DFC-LRR demonstrates a roughly linear computational speedup as a function of t, comparable accuracies to LRR for smaller values of t and a quite gradual decrease in accuracy for larger t.

4.2. Graph-based Semi-Supervised Learning

Graph representations, in which samples are vertices and weighted edges express affinity relationships between samples, are crucial in various computer vision tasks. Classical graph construction methods separately calculate the outgoing edges for each sample. This local strategy makes the graph vulnerable to contaminated data or outliers. Recent work in computer vision has illustrated the utility of global graph construction strategies using graph Laplacian [9] or matrix low-rank [32] based regularizers. L1 regularization has also been effectively used to encourage sparse graph construction [5, 13]. Building upon the success of global construction methods and noting the connection between subspace segmentation and graph construction as described in Section 2.1, we present a novel application of the low-rank representation methodology, relying on our DFC-LRR algorithm to scalably yield a *sparse, low-rank graph* (SLR-graph). We present a variety of results on large-scale semi-supervised learning visual classification tasks and provide a detailed comparison with leading baseline algorithms.

4.2.1 Benchmarking Data

We adopt the following three large-scale benchmarks:

Columbia Consumer Video (CCV) Content Detection[7]: Compiled to stimulate research on recognizing highly-diverse visual content in unconstrained videos, this dataset consists of 9317 YouTube videos over 20 semantic categories (e.g., baseball, beach, music performance). Three popular audio/visual features (5000-D SIFT, 5000-D STIP, and 4000-D MFCC) are extracted.

MED12 Multimedia Event Detection: The MED12 video corpus consists of ~150K multimedia videos, with an average duration of 2 minutes, and is used for detecting 20 specific semantic events. For each event, 130 to 367 videos are provided as positive examples, and the remainder of the videos are "null" videos that do not correspond to any event. In this work, we keep all positive examples and sample 10K null videos, resulting in a dataset of $13,876$ videos. We extract six features from each video, first at sampled frames and then accumulated to obtain video-level representations. The features are either visual (1000-D sparse-SIFT, 1000-D dense-SIFT, 1500-D color-SIFT, 5000-D STIP), audio (2000-D MFCC), or semantic features (2659-D CLASSEME [25]).

NUS-WIDE-Lite Image Tagging: NUS-WIDE is among the largest available image tagging benchmarks, consisting of over 269K crawled images from Flickr that are associated with over 5K user-provided tags. Ground-truth images are manually provided for 81 selected concept tags. We generate a lite version by sampling 20K images. For each image, 128-D wavelet texture, 225-D block-wise LAB-based color moments and 500-D bag of visual words are extracted, normalized and finally concatenated to form a single feature representation for the image.

4.2.2 Graph Construction Algorithms

The three graph construction schemes we evaluate are described below. Note that we exclude other baselines (e.g., NNLRS [32], LLE graph [28], L1-graph [5]) due to either scalability concerns or because prior work has already demonstrated inferior performance relative to the SPG algorithm defined below [32].

kNN-graph: We construct a nearest neighbor graph by connecting (via undirected edges) each vertex to its k nearest neighbors in terms of l_2 distance in the specified feature space. Exponential weights are associated with edges, i.e., $w_{ij} = \exp\left(-d_{ij}^2/\sigma^2\right)$, where d_{ij} is the distance between x_i and x_j and σ is an empirically-tuned parameter [27].

SPG: Cheng et al. [5] proposed a noise-resistant L1-graph which encourages sparse vertex connectedness, motivated by the work of sparse representation [29]. Subsequent work, entitled *sparse probability graph* (SPG) [13] enforced positive graph weights. Following the approach of [32], we implemented a variant of SPG by solving the following optimization problem for each sample:

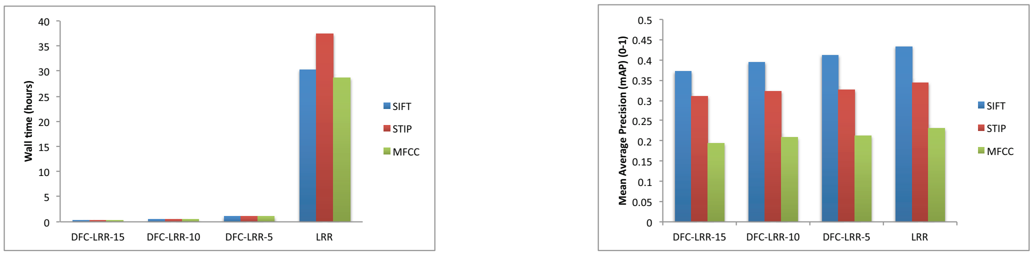
$$\min_{\mathbf{w}_x} \ \|\mathbf{x} - \mathbf{D}_x\mathbf{w}_x\|_2^2 + \alpha\|\mathbf{w}_x\|_1, \ \ s.t. \ \mathbf{w}_x \geq 0, \tag{3}$$

where $\mathbf{x}$ is a feature representation of a sample and $\mathbf{D}_x$ is the basis matrix for $\mathbf{x}$ constructed from its n_k nearest neighbors. We use an open-source tool[8] to solve this non-negative Lasso problem.

SLR-graph: Our novel graph construction method contains two-steps: first LRR or DFC-LRR is performed on the entire data set to recover the intrinsic low-rank clustering structure. We then treat the resulting low-rank coefficient matrix $\mathbf{Z}$ as an affinity matrix, and for sample x_i, the n_k samples with largest affinities to x_i are selected to form a basis matrix and used to solve the SPG optimization described by Problem (3). The resulting non-negative coefficients (typically sparse owing to the ℓ_1 regularization term on $\mathbf{w}_x$ in (3)) are used to define the graph.

[7] http://www.ee.columbia.edu/ln/dvmm/CCV/

[8] http://sparselab.stanford.edu



(a) (b)

Figure 4: Trade-off between computation and accuracy for the SLR-graph on the CCV dataset. (a) Wall time of LRR and DFC-LRR with varying numbers of subproblems. (b) mAP scores for these same experiments.

Table 1: Mean average precision (mAP) (0-1) scores for various graph construction methods. DFC-LRR-10 is performed for SLR-Graph. The best mAP score for each feature is highlighted in bold.

(a) CCV

	kNN-Graph	SPG	SLR-Graph
SIFT	.2631	.3863	**.3946**
STIP	.2011	.3036	**.3227**
MFCC	.1420	**.2129**	.2085

(b) MED12

	kNN-Graph	SPG	SLR-Graph
Color-SIFT	.0742	.1202	**.1432**
Dense-SIFT	.0928	.1350	**.1525**
Sparse-SIFT	.0780	.1258	**.1464**
MFCC	.0962	**.1371**	**.1371**
CLASSEME	.1302	.1872	**.2120**
STIP	.0620	**.0835**	.0803

(c) NUS-WIDE-Lite

kNN-Graph	SPG	SLR-Graph
.1080	.1003	**.1179**

4.2.3 Experimental Design

For each benchmarking dataset, we first construct graphs by treating sample images/videos as vertices and using the three algorithms outlined in Section 4.2.2 to create (sparse) weighted edges between vertices. For fair comparison, we use the same parameter settings, namely $\alpha = 0.05$ and $n_k = 500$ for both SPG and SLR-graph. Moreover, we set $k = 40$ for kNN-graph after tuning over the range $k = 10$ through $k = 60$.

We then use a given graph structure to perform semi-supervised label propagation using an efficient label propagation algorithm [27] that enjoys a closed-form solution and often achieves the state-of-the-art performance. We perform a separate label propagation for each category in our benchmark, i.e., we run a series of 20 binary classification label propagation experiments for CCV/MED12 and 81 experiments for NUS-WIDE-Lite. For each category, we randomly select half of the samples as training points (and use their ground truth labels for label propagation) and use the remaining half as a test set. We repeat this process 20 times for each category with different random splits. Finally, we compute Mean Average Precision (mAP) based on the results on the test sets across all runs of label propagation.

4.2.4 Experimental Results

We first performed experiments using the CCV benchmark, the smallest of our datasets, to explore the tradeoff between computation and accuracy when using DFC-LRR as part of our proposed SLR-graph. Figure 4(a) presents the time required to run SLR-graph with LRR versus DFC-LRR with three different numbers of subproblems ($t = 5, 10, 15$), while Figure 4(b) presents the corresponding accuracy results. The figures show that DFC-LRR performs comparably to LRR for smaller values of t, and performance gradually degrades for larger t. Moreover, DFC-LRR is up to two orders of magnitude faster and achieves superlinear speedups relative to LRR.[9] Given the scalability issues of LRR on this modest-sized dataset, along with the comparable accuracy of DFC-LRR, we ran SLR-graph exclusively with DFC-LRR ($t = 10$) for our two larger datasets.

Table 1 summarizes the results of our semi-supervised learning experiments using the three graph construction techniques defined in Section 4.2.2. The results show that our proposed SLR-graph approach leads to significant performance gains in terms of mAP across all benchmarking datasets for the vast majority of features. These results demonstrate the benefit of enforcing both low-rankedness and sparsity during graph construction. Moreover, conventional low-rank oriented algorithms, e.g., [32, 16] would be computationally infeasible on our benchmarking datasets,

[9] We restricted the maximum number of internal LRR iterations to 500 to ensure that LRR ran to completion in less than two days.

thus highlighting the utility of employing DFC's divide-and-conquer approach to generate a scalable algorithm.

5. Conclusion

Our primary goal in this work was to introduce a provably accurate algorithm suitable for large-scale low-rank subspace segmentation. While some contemporaneous work [1] also aims at scalable subspace segmentation, this method offers no guarantee of correctness. In contrast, DFC-LRR provably preserves the theoretical recovery guarantees of the LRR program. Moreover, our divide-and-conquer approach achieves empirical accuracy comparable to state-of-the-art methods while obtaining linear to superlinear computational gains, both on standard subspace segmentation tasks and on novel applications to semi-supervised learning. DFC-LRR also lays the groundwork for scaling up LRR derivatives known to offer improved performance, e.g., LatLRR in the setting of standard subspace segmentation and NNLRS in the graph-based semi-supervised learning setting. The same techniques may prove useful in developing scalable approximations to other convex formulations for subspace segmentation, e.g., [20].

References

[1] A. Adler, M. Elad, and Y. Hel-Or. Probabilistic subspace clustering via sparse representations. *IEEE Signal Process. Lett.*, 20(1):63–66, 2013. 8

[2] R. Basri and D. Jacobs. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(3):218–233, 2003. 1

[3] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, 2011. 5

[4] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan. Multi-task low-rank affinity pursuit for image segmentation. In *ICCV*, 2011. 2

[5] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. S. Huang. Learning with l1-graph for image analysis. *IEEE Transactions on Image Processing*, 19(4):858–866, 2010. 6

[6] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3), 1998. 1

[7] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *CVPR*, 2009. 1

[8] B. R. Feng Niu, C. Ré, and S. J. Wright. Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. In *NIPS*, 2011. 1

[9] S. Gao, I. W. H. Tsang, L. T. Chia, and P. Zhao. Local features are not lonely - laplacian sparse coding for image classification. In *CVPR*, 2010. 6

[10] C. W. Gear. Multibody grouping from motion images. *Int. J. Comput. Vision*, 29:133–150, August 1998. 1, 2

[11] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis. Large-scale matrix factorization with distributed stochastic gradient descent. In *KDD*, 2011. 1

[12] T. Hastie and P. Simard. Metrics and models for handwritten character recognition. *Statistical Science*, 13(1):54–65, 1998. 1

[13] R. He, W.-S. Zheng, B.-G. Hu, and X.-W. Kong. Nonnegative sparse coding for discriminative semi-supervised learning. In *CVPR*, 2011. 6

[14] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. 10

[15] S. Kumar, M. Mohri, and A. Talwalkar. On sampling-based approximate spectral decomposition. In *ICML*, 2009. 3

[16] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. UIUC Technical Report UILU-ENG-09-2214, 2009. 7

[17] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *ICML*, 2010. 1, 2, 4, 5

[18] G. Liu, H. Xu, and S. Yan. Exact subspace segmentation and outlier detection by low-rank representation. `arXiv:1109.1646v2[cs.IT]`, 2011. 1, 2, 3, 4, 5, 9, 10, 11

[19] G. Liu and S. Yan. Latent low-rank representation for subspace segmentation and feature extraction. In *ICCV*, 2011. 1

[20] G. Liu and S. Yan. Active subspace: Toward scalable low-rank learning. *Neural Computation*, 24:3371–3394, 2012. 8

[21] L. Mackey, A. Talwalkar, and M. I. Jordan. Divide-and-conquer matrix factorization. In *NIPS*, 2011. 1, 2, 3, 4, 9

[22] B. Recht. A simpler approach to matrix completion. `arXiv:0910.0651v2[cs.IT]`, 2009. 3

[23] B. Recht and C. Ré. Parallel stochastic gradient algorithms for large-scale matrix completion. In *Optimization Online*, 2011. 1

[24] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography. *International Journal of Computer Vision*, 9(2):137–154, 1992. 1

[25] L. Torresani, M. Szummer, and A. W. Fitzgibbon. Efficient object category recognition using classemes. In *ECCV*, 2010. 6

[26] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (gpca). *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(12):1–15, 2005. 1

[27] F. Wang and C. Zhang. Label propagation through linear neighborhoods. In *ICML*, 2006. 6, 7

[28] J. Wang, F. Wang, C. Zhang, H. C. Shen, and L. Quan. Linear neighborhood propagation and its applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(9):1600–1615, 2009. 6

[29] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(2):210–227, 2009. 6

[30] A. Yang, J. Wright, Y. Ma, and S. Sastry. Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding*, 110(2):212–225, 2008. 1

[31] H.-F. Yu, C.-J. Hsieh, S. Si, and I. Dhillon. Scalable coordinate descent approaches to parallel matrix factorization for recommender systems. In *ICDM*, 2012. 1

[32] L. Zhuang, H. Gao, Z. Lin, Y. Ma, X. Zhang, and N. Yu. Non-negative low rank and sparse graph for semi-supervised learning. In *CVPR*, 2012. 1, 2, 6, 7