Robust Dictionary Learning with Capped $\ell_1$-Norm

Wenhao Jiang, Feiping Nie, Heng Huang*
University of Texas at Arlington
Arlington, Texas 76019, USA
cswhjiang@gmail.com, feipingnie@gmail.com, heng@uta.edu

Abstract
Expressing data vectors as sparse linear combinations of basis elements (dictionary) is widely used in machine learning, signal processing, and statistics. It has been found that dictionaries learned from data are more effective than off-the-shelf ones. Dictionary learning has become an important tool for computer vision. Traditional dictionary learning methods use quadratic loss function which is known sensitive to outliers. Hence they could not learn the good dictionaries when outliers exist. In this paper, aiming at learning dictionaries resistant to outliers, we proposed capped $\ell_1$-norm based dictionary learning and an efficient iterative re-weighted algorithm to solve the problem. We provided theoretical analysis and carried out extensive experiments on real word datasets and synthetic datasets to show the effectiveness of our method.

1 Introduction
Dictionary learning [Tosic and Frossard, 2011] and sparse representation [Olshausen and Field, 1997] are important tools for computer vision. Dictionary learning seeks to learn an adaptive set of basis elements (dictionary) from data instead of predefined ones [Mallat, 1999], so that each data sample is represented by sparse linear combination of these basis vectors. It has achieved state-of-the-art performance for numerous image processing tasks such as classification [Raina et al., 2007; Mairal et al., 2009b], denoising [Elad and Aharon, 2006], self-taught learning [Wang et al., 2013b], and audio processing [Grosse et al., 2007]. Unlike principal component analysis, dictionary learning does not impose that the dictionary be orthogonal, hence allow more flexibility to represent data.

Usually, dictionary learning is formulated as:

$$\min_{\mathbf{D} \in \mathbb{C}^d \times K} \| \mathbf{X} - \mathbf{D} \mathbf{A} \|^2_F, \quad s.t. \quad \| \mathbf{A} \|_0 \leq \gamma. \quad (1)$$

In this formulation, $\mathbf{X} \in \mathbb{R}^{d \times n}$ is the data matrix whose columns represent samples, $\mathbf{A} \in \mathbb{R}^{K \times n}$ is the new representations of data and the matrix $\mathbf{D} \in \mathbb{R}^{d \times K}$ contains $K$ basis vectors to learn. To prevent the $\ell_2$ norm of $\mathbf{D}$ being arbitrarily large which would lead to arbitrarily small values in the columns of $\mathbf{A}$, it is common to constraint the columns have $\ell_2$ norm less than or equal to 1. Hence, $\mathbf{D}$ is in the following convex set of matrices:

$$\mathcal{C} = \{ \mathbf{D} \in \mathbb{R}^{d \times K} \mid \forall i \in \{1, \cdots, K\}, \| \mathbf{d}_i \|_2 \leq 1 \}. \quad (2)$$

The dictionary learned is the foundation for sparse representations. Dictionary with good expressive ability is the key to achieve good performance. It is well known that the quadratic loss function is not robust to outliers. To train the dictionary, a large amount data will be used, and outliers will be included in the training data unavoidably. Hence it is necessary to learn dictionary resistant to outliers.

To be robust to outliers, the quadratic loss function could be replaced by an loss function that are not sensitive to outliers, e.g. $\ell_1$ norm loss or Huber loss. In this paper, we use capped $\ell_1$-norm based loss function $l_u(\mathbf{v}) = \min(|\mathbf{v}|, \varepsilon)$, where $\varepsilon$ is a parameter. It is illustrated in Fig. 1. This loss function treat $\mathbf{v}$ equally if $|\mathbf{v}|$ is bigger than $\varepsilon$. Hence, it is more robust to outliers than $\ell_1$ norm. Unfortunately, this loss function is not convex. In this paper, we propose an efficient algorithm to find the local optimal solutions. And we will also provide theoretical analysis of this method.

2 Related Work
In this section, we present a brief review on dictionary learning methods as follows.

Since the original dictionary learning model (1) is NP-hard, the straightforward way to find dictionary is to adopt greedy
strategy. K-SVD [Aharon et al., 2006] tried to solve the following model:

\[
\min_{D,A} \frac{1}{2} \|X - DA\|_2^2 \\
\text{s.t. } d_j = 1, \text{ for } j = 1, 2, \ldots, K \\
\|a_i\|_0 \leq T_0, \text{ for } i = 1, 2, \ldots, n,
\]

where \(d_i\) is the \(i\)th column of matrix \(D\). The K-SVD is an iterative method that alternates between sparse coding of the examples based on the current dictionary and a process of updating the dictionary atoms. The columns of dictionary are updated with SVD sequentially, and the corresponding coefficients are updated with any pursuit algorithm. The K-SVD algorithm showed good performance on image denoising. As extensions of K-SVD, LC-KSVD [Jiang et al., 2013] and discriminative K-SVD [Zhang and Li, 2010] introduced label information into the procedure of learning dictionaries. Hence the dictionaries learned are more discriminative.

Except for adopting greedy strategy to solve problems with \(\ell_0\) constraints, the original problem can be relaxed into the following traditional dictionary learning problem with \(\ell_1\) regularization:

\[
\min_{D \in \mathbb{C}^{n \times K}, A} \|X - DA\|_F^2 + \lambda \|A\|_1.
\]

(4)

It is convex separately with respect to \(D\) and \(A\), a \(\ell_1\) regularized least squares problem and a \(\ell_2\) constrained least squares problem are needed to be solved iteratively to find the solutions. In [Lee et al., 2007], an feature-sign search algorithm for learning coefficients, and a Lagrange dual method for learning the bases are proposed to compute the dictionary and corresponding representations. An online algorithm was proposed to solve it efficiently in [Mairal et al., 2009a]. The method mentioned above all use quadratic loss functions to measure the reconstruction errors, hence these methods are not robust enough to outliers.

In order to learn dictionary robustly, Lu et al. proposed online robust dictionary learning (ORDL) [Lu et al., 2013], which uses \(\ell_1\) loss function and \(\ell_1\) regularization term that could be expressed as

\[
\min_{D,A, \|d\|_1 \leq 1} \sum_{i=1}^{n} \|x_i - Da_i\|_1^2 + \lambda \|a\|_1.
\]

(5)

It updates dictionary and representations alternately in an online way. For visual tracking, Wang proposed online robust non-negative dictionary learning [Wang et al., 2013c], which uses the Huber loss function. In [Wang et al., 2013a], the semi-supervised robust dictionary learning model was proposed by solving the \(\ell_p\)-norm based objective.

3 Proposed New Algorithm

To facilitate the presentation of our method, we describe the notations in the following subsection.

3.1 Notations

Throughout this paper, the following definitions and notations are used. We use bold upper letters for matrices, bold lower letters for vectors and regular lower letters for elements. For vector \(a = (a_1, a_2, \ldots, a_m)^T \in \mathbb{R}^m\), let \(\|a\|_1 = \sum_{i=1}^{m} |a_i|\) be the \(\ell_1\)-norm of \(a\). Similarly, we define \(\|A\|_1\) as the number of nonzero elements in matrix \(A\) and \(\|A\|_1 = \sum \|a\|_1\) be the \(\ell_1\) norm of matrix \(A\).

3.2 Robust dictionary learning

Capped \(\ell_1\)-norm has been used in [Zhang, 2010; 2013] as a better approximation of \(\ell_0\)-norm regularization. An extension capped-\(\ell_{1,1}\) regularization was proposed in the multi-task feature learning setting [Gong et al., 2013]. In this paper, we use capped \(\ell_1\)-norm as a robust loss function.

For dictionary learning, our goal is to find a set of high quality atoms. An straightforward way is to use \(\ell_1\) loss function. To provide better robustness, we go further to use capped \(\ell_1\)-norm loss function. As illustrated in Fig 1, the objective values of capped \(\ell_1\)-norm loss does not increase any more if \(|u|\) is larger than \(\epsilon\). Therefore, capped \(\ell_1\)-norm loss is more robust than \(\ell_1\)-norm loss. We use the same constraints for dictionary and the objective function of our robust dictionary learning with capped \(\ell_1\)-norm is expressed as:

\[
\min_{D,A, \|d\|_1 \leq 1} \sum_{i=1}^{n} \min(||x_i - Da_i||_2, \epsilon) + \lambda ||A||_1.
\]

(6)

From this objective function, we can see that \(\epsilon\) is set as \(\infty\), the above objective function becomes \(\ell_{2,1}\)-norm with the same constraints.

We define \(f(A) = ||A||_1\), which is a convex function. And define a concave function \(L(u) = \min(\sqrt{u}, \epsilon)\) where \(u > 0\) and \(g(D,a) = ||x - Da||_2^2\). We have:

\[
L'(u) = \begin{cases} 
\frac{1}{2\sqrt{u}}, & \text{if } 0 < u < \epsilon^2 \\
0, & \text{if } u > \epsilon^2
\end{cases}
\]

(7)

Our objective function is sum of a convex function and a concave function. According to the re-weighted method proposed in [Nie et al., 2010; 2014], it can be solved by updating \(D, A\) and auxiliary variables \(s_i\) with following updating rules:

\[
[D,A] = \arg \min_{D,A, \|d\|_1 \leq 1} \sum_{i=1}^{n} s_i ||x_i - Da_i||_2^2 + \lambda ||A||_1.
\]

(8)

\[
s_i = \begin{cases} 
\frac{1}{2||x_i - Da_i||_2}, & \text{if } ||x_i - Da_i||_2 \leq \epsilon \\
0, & \text{otherwise}
\end{cases}
\]

(9)

The whole iterative re-weighted method is listed in Alg. 1.

The subproblem (8) is similar to the traditional dictionary learning. The only difference lies in the fact that samples are not treated equally. It is weighted dictionary learning. From the updating rule for \(s_i\), we can see that the samples with lower reconstruction errors have higher weights, which is very intuitive for learning dictionary robustly.

The subproblem (8) is convex separately with respect to \(D\) and \(A\). We adopt the same strategy as solving traditional dictionary learning. We solve it by updating \(D\) and \(A\) alternately. With \(A\) fixed, the objective function becomes:

\[
\min_{D, \|d\|_1 \leq 1} \sum_{i=1}^{n} s_i ||x_i - Da_i||_2^2,
\]

(10)
which is equivalent to:

$$
\min_{D, \|d_i\| \leq 1} \sum_i \|z_i - Dc_i\|^2, \quad (11)
$$

where $z_i = \sqrt{\tau_i}x_i$ and $c_i = \sqrt{\tau_i}a_i$. To solve the above problem, we update $D$ column by column which is similar to the traditional dictionary learning. The algorithm we use is adopted from [Mairal et al., 2009a], which is described in Alg. 3.

With $D$ fixed, the objective of (8) becomes:

$$
A = \arg\min_A \sum_i s_i \|x_i - Da_i\|^2 + \lambda \|A\|_1, \quad (12)
$$

which can be decomposed into $n$ independent problems as follows:

$$
a_i = \arg\min_a s_i \|x_i - Da_i\|^2 + \lambda \|a\|_1. \quad (13)
$$

If $s_i \neq 0$, this problem can be transformed into the following LASSO problem:

$$
a_i = \arg\min_a \|x_i - Da_i\|^2 + \frac{\lambda}{s_i} \|a\|_1, \quad (14)
$$

which could be solved efficiently. The algorithm to solve weighted dictionary learning is summarized in Alg. 2.

During the stage of training dictionary $D$, if $s_i$ is 0, the corresponding representation $a_i$ will be a zero vector. Hence, when training $D$, our method does not find codings for outliers. However, the outliers in training $D$ stage might not be outliers for classification tasks. Hence, we let the classification algorithm to decide how to use the codings of the outliers. With the learned $D$, in order to get the codings for outliers, we just run Alg. 1 with setting $\varepsilon$ as $\infty$ and omit the $D$ updating step (step 1 in Alg. 2) to learn the representations for both training data and testing data with the same $\lambda$. In fact, we are solving a dictionary learning problem with $\ell_{2,1}$-norm loss function.

Algorithm 1 Robust dictionary learning with capped $\ell_1$-norm

**input:** Data matrix $X$, dictionary size $K$, $\lambda$ and $\varepsilon$.

**Initialize** $D, A$ and $s_i = 1$ for $i = 1, 2, \cdots, n$.

**repeat**

1. Solve subproblem (8) with algorithm 2
2. Update $s_i$ for $i = 1, \cdots, n$ as (9)

**until** Converge

**output:** $D$ and $A$

Algorithm 2 Weighted dictionary learning

**input:** Data matrix $X$, initial dictionary matrix $D_0$ and representation matrix $A_0$, $\lambda$ and weight vector $s$.

**Initialize** $D = D_0$ and $A = A_0$

**repeat**

1. Update $D$ with algorithm 3.
2. Update $a_i$ for $i = 1, \cdots, n$ as

$$
a_i = \begin{cases} 
\arg\min_a \|x_i - Da_i\|^2 + \frac{\lambda}{s_i} \|a\|_1 & \text{if } s_i > 0 \\
0 & \text{otherwise}
\end{cases}
$$

**until** Converge

**output:** $D, A$

Algorithm 3 Dictionary update

**input:** Data matrix $X$, dictionary matrix $D$, representation matrix $A$ and weight vector $s$.

**Compute** matrix $G = [g_1, \cdots, g_K] \in \mathbb{R}^{K \times K}$ and $H = [h_1, \cdots, h_K] \in \mathbb{R}^{d \times K}$ as

$$
G = \sum_{i=1}^n s_i a_i a_i^T, \quad H = \sum_{i=1}^n s_i x_i a_i^T.
$$

**repeat**

for $j = 1$ to $K$

Update the $j$-th column of $D$:

$$
u_j = \frac{1}{G_{jj}} (h_j - Dg_j) + d_j,
$$

$$
d_j = \frac{1}{\max(\|u_j\|_2, 1)} u_j
$$

**end for**

**until** Converge

**output:** $D$

where $L^*(s)$ is the concave dual of $\tilde{L}(u)$ defined as:

$$
L^*(s) = \inf_u \left[ su - \tilde{L}(u) \right], \quad (17)
$$

Plug in $\tilde{L}(u)$ and it is easy to find that:

$$
L^*(s) = \begin{cases}
-\frac{1}{\varepsilon}, & \text{if } \sqrt{s} < \varepsilon \\
0, & \text{if } \sqrt{s} \geq \varepsilon
\end{cases}.
$$

Therefore, the capped $\ell_1$-norm loss could be expressed as:

$$
\min_{s \geq 0} \|x - Da\|_2, \varepsilon = \inf_{s \geq 0} L_x(s, D, a), \quad (19)
$$

where

$$
L_x(s, D, a)
$$

$$
= \begin{cases}
\frac{s}{2} \|x - Da\|_2^2 + \frac{1}{2\varepsilon}, & \text{if } \|x - Da\|_2 < \varepsilon \\
\frac{s}{2} \|x - Da\|_2^2 - s\varepsilon^2 + \varepsilon, & \text{if } \|x - Da\|_2 \geq \varepsilon
\end{cases}.
$$

Hence, the objective function (6) is equivalent to the following objective:

$$
\min_{D, A, \|a_i\| \leq 1} \sum_{i=1}^n \inf_{s_i \geq 0} L_x(s_i, D, a_i) + \lambda \|a_i\|_1, \quad (21)
$$

3.3 Convergence analysis

Theorem 3.1. The algorithm described in Alg. 1 will decrease the objective value of (6) in each iteration until it converges.

Proof. We define $h(u) = u^2$, and denote $u = \|x - Da\|_2^2$ and $\tilde{L}(u) = \min(\sqrt{u}, \varepsilon)$. The capped $\ell_1$-norm loss function $\min(\|x - Da\|_2, \varepsilon)$ could be re-written as:

$$
\min(\|x - Da\|_2, \varepsilon) = \inf_{s \geq 0} [sh(\|x - Da\|_2) - L^*(s)], \quad (15)
$$

$$
L^*(s) = \inf_u \left[ su - \tilde{L}(u) \right]. \quad (17)
$$

Plug in $\tilde{L}(u)$ and it is easy to find that:

$$
L^*(s) = \begin{cases}
-\frac{1}{\varepsilon}, & \text{if } \sqrt{u} < \varepsilon \\
0, & \text{if } \sqrt{u} \geq \varepsilon
\end{cases}. \quad (18)
$$

Therefore, the capped $\ell_1$-norm loss could be expressed as:

$$
\min_{s \geq 0} \|x - Da\|_2, \varepsilon = \inf_{s \geq 0} L_x(s, D, a), \quad (19)
$$

where

$$
L_x(s, D, a)
$$

$$
= \begin{cases}
\frac{s}{2} \|x - Da\|_2^2 + \frac{1}{2\varepsilon}, & \text{if } \|x - Da\|_2 < \varepsilon \\
\frac{s}{2} \|x - Da\|_2^2 - s\varepsilon^2 + \varepsilon, & \text{if } \|x - Da\|_2 \geq \varepsilon
\end{cases}. \quad (20)
$$

Hence, the objective function (6) is equivalent to the following objective:

$$
\min_{D, A, \|a_i\| \leq 1} \sum_{i=1}^n \inf_{s_i \geq 0} L_x(s_i, D, a_i) + \lambda \|a_i\|_1, \quad (21)
$$

3592
which can be re-written as:

\[
\min_{D, A, \|d_i\| \leq 1, s_i \geq 0} o(D, A, s),
\]

where we denote:

\[
o(D, A, s) = \sum_{i=1}^{n} L_{x_i}(s_i, D, a_i) + \lambda \|a_i\|_1.
\]

Therefore, the algorithm described in Alg. 1 can be seen as a two stage optimization method:

**Updating D and A stage:** D and A are updated by solving:

\[
\min_{D, A, \|d_i\| \leq 1} o(D, A, s),
\]

which is equivalent to:

\[
\min_{D, A, \|d_i\| \leq 1} \sum_{i=1}^{n} s_i \|x_i - Da_i\|_2^2 + \lambda \|a_i\|_1.
\]

The updated D and A decrease the objective values of (25). Hence, they also decrease the objective values of (24).

**Updating s_i stage:** The auxiliary variable s are updated by solving:

\[
\min_{s_i \geq 0} o(D, A, s),
\]

which is equivalent to n independent problems:

\[
\min_{s_i \geq 0} L_{x_i}(s_i, D, a_i).
\]

Both stages will decrease the value of the objective function (21), hence our algorithm also decrease the value of the original objective function (6) and is guaranteed to converge.

### 4 Experimental Results

In this section, we analyze and illustrate the performance of our method on real word datasets.

#### 4.1 Preliminary study on natural image patches

First, we present a preliminary analysis of our method on natural image patches. In our method, \( \lambda \) mainly controls the sparsity of representations and \( \varepsilon \) mainly controls the number of outliers identified by the algorithm. \( \varepsilon \) is related to the residuals of representations. If the residual of a sample is larger than \( \varepsilon \), it is not used to train the dictionary, since the corresponding \( s \) is zero. We set \( \varepsilon \) as infinity first to get the distribution of the residuals. We use this distribution as an estimation of the distribution of residuals with optimal D and A. We set \( \varepsilon \) such that a fraction of samples are seen as outliers. In this experiment, we set the fraction of outliers as 0.05 and \( \lambda = 0.1 \) empirically. The values of objective function during iterations are presented in Fig. 2. We can see that our method converged in only 22 iterations. The bases learned on natural image patches by our method is shown in Fig. 3.

### 4.2 Face recognition without occlusion

In this subsection, we provide the experimental results on face recognition tasks without synthetic occlusion on extended Yale B dataset [Georghiades et al., 2001] and AR face dataset [Martinez, 1998].

**Extended Yale B dataset** The extended Yale B database [Georghiades et al., 2001] contains 2,414 images of 38 human frontal faces under 64 illumination conditions and expressions. There are about 64 images for each person. The original images were cropped to 192 \( \times \) 168 pixels. Some samples are shown in Fig. 4(a). We do not perform any pre-process on the images. Following [Wright et al., 2009], we project each face image into a 504-dimensional feature vector using a random matrix. We split the database randomly into two halves. One half which contains about 32 images for each person was used for training the dictionary. The other half was used for testing.
In the practical implementation of our method, we first set \( \varepsilon \) as infinity and choose the best \( \lambda \) with cross validation. Then we fix \( \lambda \) and choose \( \varepsilon \).

For face recognition, we compared our method with a few start-of-the-art methods, including traditional dictionary learning [Mairal et al., 2009a], K-SVD [Aharon et al., 2006], LC-KSVD1 and LC-KSVD2 [Jiang et al., 2013], D-KSVD [Zhang and Li, 2010] and ORDL [Lu et al., 2013]. The dictionary size is 570 for all methods, which means 15 items for person on average. The parameters for these methods were selected by cross validation. We ran all methods on 10 different splits of training and testing set. The results are summarized in Table 1. We can see that our method achieved the best performance. The reason that ORDL did not perform well might be the dictionary and representations trained with \( \ell_1 \)-norm loss function are not suitable for classification tasks.

We show that our method is robust when training dictionary. Recall that, in our method \( s_i = 0 \) means that the \( i \)th sample is recognized as outlier and not used in training the dictionary. In running on one random split, our method found 20 outliers, which are all shown in Fig. 5. We can see that most of the outliers found by our method are with extreme illumination, which will effect the quality of bases.

**AR face dataset** The AR face dataset [Martinez, 1998] consists of over 4,000 color images from 126 individuals. For each individual, 26 pictures were taken in two separate sessions. Compared to the extended Yale B dataset, as shown in Fig. 4(b) \(^1\), the AR face dataset includes more facial variations, including different illumination conditions, different expressions, and different facial disguises (sunglasses and scarves). Following the standard evaluation procedure from [Wright et al., 2009], we used a subset of the database consisting of 2,600 images from 50 male subjects and 50 female subjects. For each person, 20 images were randomly selected for training and the remaining images are for testing. Each face image is cropped to \( 165 \times 120 \) and then projected into a 540-dimensional feature vector.

Similar to the experiments on Yale B dataset, we also ran all methods in 10 different splits of training and testing set. The results are presented in Table 2. We can see that our method achieved the best performance among all dictionary learning algorithms. We show the 17 outliers found by our algorithm in Fig 6. We can see that these outliers are faces with glasses or scarves. Our algorithm identified these faces as outliers that will hurt the quality of dictionary and did not use them in the training process.

### 4.3 Face recognition with occlusion

In order to study the property of robustness to outliers, we carried out experiments on extended Yale B dataset with syn-

---

\(^1\)We use grayscale images in our experiments.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capped Norm</td>
<td><strong>96.91</strong></td>
</tr>
<tr>
<td>Traditional DL</td>
<td>95.70</td>
</tr>
<tr>
<td>KSVD</td>
<td>95.54</td>
</tr>
<tr>
<td>LC-KSVD1</td>
<td>93.61</td>
</tr>
<tr>
<td>LC-KSVD2</td>
<td>94.48</td>
</tr>
<tr>
<td>D-KSVD</td>
<td>93.58</td>
</tr>
<tr>
<td>ORDL</td>
<td>89.17</td>
</tr>
</tbody>
</table>

---

**Table 1:** Average classification accuracies(%) on extended Yale B dataset.
<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th>Method</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capped Norm</td>
<td>97.48</td>
<td>Traditional DL</td>
<td>97.25</td>
</tr>
<tr>
<td>KSVD</td>
<td>95.03</td>
<td>KSVD</td>
<td>94.56</td>
</tr>
<tr>
<td>LC-KSVD1</td>
<td>94.33</td>
<td>LC-KSVD2</td>
<td>94.33</td>
</tr>
<tr>
<td>D-KSVD</td>
<td>88.18</td>
<td>ORDL</td>
<td>91.72</td>
</tr>
</tbody>
</table>

Table 2: Average classification accuracies(%) on AR face dataset.

Figure 6: The 17 outliers found by our method when training dictionary on an randomly selected training set from AR face dataset.

Table 3: Average classification accuracies(%) on extended Yale B dataset with block occlusion of different levels.

<table>
<thead>
<tr>
<th>Method</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capped Norm</td>
<td>96.29</td>
<td>95.49</td>
<td>95.43</td>
<td>93.88</td>
</tr>
<tr>
<td>Traditional DL</td>
<td>90.00</td>
<td>88.37</td>
<td>87.61</td>
<td>86.17</td>
</tr>
<tr>
<td>KSVD</td>
<td>93.74</td>
<td>93.26</td>
<td>92.99</td>
<td>92.30</td>
</tr>
<tr>
<td>LC-KSVD1</td>
<td>94.10</td>
<td>93.47</td>
<td>93.14</td>
<td>92.42</td>
</tr>
<tr>
<td>LC-KSVD2</td>
<td>94.27</td>
<td>93.65</td>
<td>93.21</td>
<td>92.27</td>
</tr>
<tr>
<td>D-KSVD</td>
<td>91.19</td>
<td>90.14</td>
<td>89.85</td>
<td>89.21</td>
</tr>
<tr>
<td>ORDL</td>
<td>87.00</td>
<td>85.53</td>
<td>84.95</td>
<td>83.38</td>
</tr>
</tbody>
</table>

and are difficult to recognize. In our method, the dictionaries were trained without these samples, hence they were better than dictionaries trained in other ways, which can be proved from the comparisons of performances in Table 3.

Figure 8: The 93 outliers found by our method when training dictionary on an randomly selected training set with 10% samples corrupted with block occlusion from extended Yale B dataset.

5 Conclusion

In this paper, we presented a robust dictionary learning model based on capped $\ell_1$-norm and an efficient algorithm to find local solutions. One important advantage of our method is its robustness to outliers. By iteratively assigning weights to samples, our algorithm succeeded in finding outliers and reduced their effects on training the dictionaries. The proposed method was extensively evaluated on face recognition jobs on different real word datasets and synthetic datasets. The experimental results demonstrated that our method outperforms previous state-of-the-art dictionary learning methods.
References


