Sparse Affinity Propagation for Image Analysis

Xue Zhang, Jian Cheng Lv
Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu 610065, P.R. China.
Email: yimizx@gmail.com, lvjiancheng@scu.edu.cn

***Abstract*— It is challenging to find a small set of data points, so-called exemplars or landmarks, that are nicely representative of itself and other data points. Affinity propagation (AP) is an effective algorithm that identifies exemplars among data points by recursively sending real-valued messages between pairs of data points. AP calculates the message using the similarity among data points. Hence, the construction of similarity graph lies in the heart of the AP algorithm. A common choice for similarity is negative Euclidean distance. However, most data points, especial high-dimensional data, lies into the non-Euclidean space such that Euclidean distance cannot capture the real structure of data. Moreover, Euclidean distance is sensitive to noise and outliers such that the performance of the algorithm will be degraded when data are grossly corrupted. In this paper, we propose an algorithm, named as Sparse Affinity Propagation (SAP), which adopts sparse representation coefficient to depict the relationship among data points. For a given data set, SAP calculates the sparse representation for each data point by solving a convex problem; and then, builds a similarity graph using the representation coefficient; after that, obtains the exemplars by performing AP over the sparse similarity graph. To verify the efficacy of our algorithm, we carried out numerous experiments in the context of data summarization. Empirical studies show that SAP is superior to AP and other baseline algorithms for image analysis in accuracy and robustness.**

***Index Terms*— Data Summarization, Subset Selection, ℓ_1-minimization, Compressive Sensing, Message Propagation.**

I. INTRODUCTION

AUTOMATIC data summarization, which attempts to choose a set of representative data points (so-called exemplars, landmarks, or representatives) to depict the whole data set, is useful in massive data analysis such as keywords extraction, video/document summarization, information retrieval. Moreover, Finding Exemplars could obviously dramatically reduce the computational cost for classification. For example, one could classify the testing sample by comparing with k exemplars but with all n training samples, where $k \ll n$. Therefore, it has motivated a lot of works in the area of machine learning and pattern recognition.

To select representative points from the data set, several works have been proposed based on different assumptions. k-medoids [1], which is a variant of k-means clustering algorithm, assumes that the data distributed around the clustering centers (i.e., exemplars). It iteratively finds the centers from the data set by minimizing the distance between points labeled to be in a cluster and a point designated as the center of that cluster. When similarity/dissimilarity between pairs of data points are given, the performance of k-medoids largely depends on the initialization. The Rank Revealing QR (RRQR) [2], [3] assumes that the data have low-rank structure and selects a well-conditioned sub-matrix that spans the range of the data matrix. Moreover, some works [4]–[9] assume that the data can be represented as a linear combination of the exemplars, and then finding exemplars are formulated as a sparse recovery problem.

Moreover, Affinity Propagation (AP) algorithm [10], [11] identifies the exemplars among data points by simultaneously considering all data point as potential exemplars and exchanging messages between data points until a set of exemplars are founded. The algorithm takes a collection of real-valued similarities between pairs of data points as input to calculate $r(i,k)$ and $a(k,i)$, where $r(i,k)$ (so-called responsibility) denotes how well the kth data point is suited to be the exemplar for data point i. Moreover, AP assigns a real number $a(k,k)$ (so-called availability, the accumulated evidence of $a(k,i)$) for each data point k so that the data point with larger value of $a(k,k)$ is more likely to be chosen as exemplar. In general, AP takes the negative Euclidean distance as similarity metric, i.e., for points x_i and x_j, $s(i,j) = -\|x_i - x_j\|^2$. By iteratively maximizing responsibility and minimizing availability for each data point, AP finds a collection of exemplars and groups the data points around these exemplars. One of main attraction of AP is that it doesn't requires to pre-specify the number of exemplars, which is very useful in real applications.

AP is derived as an application of the max-sum algorithm in a factor graph, i.e., it seeks for the minima of an energy function on the basis of message passing between pairs of data points. The performance of AP strongly depends on the similarity metric. A common choice for the similarity is negative Euclidean distance. With the distance metric, AP has achieved impressive results in attribute reduction [12], document analysis [13], gene detection [14], [15], and recommendation system [16]. In some applications, however, the data (e.g., facial images) doesn't lie into the Euclidean space such that Euclidean distance could not capture the real relationship among data points. Furthermore, Euclidean distance has some demerits as the other pairwise metrics. Specifically, the

Manuscript received ***. © 2005 Journal of Software.
This work was supported by Specialized Research Fund for the Doctoral Program of Higher Education under Grant 2010081110053.

doi:10.4304/jsw.9.3.748-756

similarity based on pairwise distance is independent from the other points. Thus, the metric could not reflect the data distribution, which results in indistinctive neighborhood structure for each datum. On the other hand, the similarity based on pairwise distance is very sensitive to data noise such that it might be failed when data contains noises or outliers. Indeed, the data noises are inevitable especially for multimedia data. To address the problems, several works have been proposed [17], [18].

In this paper, we present an algorithm, named as Sparse Affinity Propagation (SAP), which takes sparse representation coefficient as similarity for the AP algorithm. For a given data set, SAP calculates the sparse representation for each datum by solving a convex optimization problem; and then, constructs a similarity graph using the representation; after that, performs the AP algorithm over the graph to obtain the exemplars.

SAP owns the merits of sparse representation and the AP algorithm. Sparse representation aims to find the minimal entries to represent each datum, which doesn't consider the global structure of data set. Hence, the optimization program of sparse representation might not produce a compact collection of data points. By performing AP over the similarity graph based on sparse representation, SAP actually eliminates the redundancy of sparse representation caused by transmissibility of linear encoding. On the other hand, SAP defines the the responsibility and availability using the sparse representation, which makes SAP is more robust than AP since the similarity between any two data points depends on not only themselves but also the other points. Moreover, SAP doesn't require specifying the size of neighborhood for each datum, and it discovers the neighbors for each point by utilizing the intrinsic sparsity. Thus, SAP has datum-adaptive neighborhood.

The rest of the paper is organized as follows: Section II presents k-medoids algorithm and affinity propagation algorithm (AP); Section III presents our Sparse Affinity Propagation (SAP). Section IV carries out some experiments to examine the effectiveness of the algorithm by classifying testing samples using the exemplars. Finally, Section V summaries this work.

II. Preliminaries

For a given data set $\mathcal{X} = \{x_1, x_2, \cdots, x_n\}$ with m dimensionality, AP automatically finds the exemplars from $\mathcal{X}$ by considering all data points as possible exemplars and exchanging real-valued message between them until the algorithm converges (as shown in Fig. 1). The messages are calculated using the negative Euclidean distance among data points and are updated on the basis of a simple rule that reflects sum-product or max-product update rules. At any time, the magnitude in each message reflects the possibility that the corresponding point has been chosen as an exemplar.

Algorithm 1 shows how does AP identify exemplars from a given data set. At the beginning, AP sets the value of $r(i,k)$ and $a(k,i)$ as 0, and considers all data points as

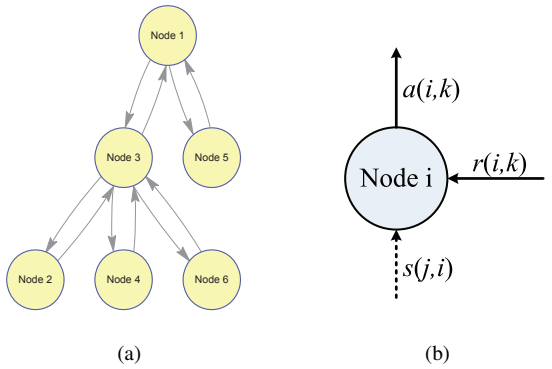


Figure 1. (a) AP is an algorithm for finding exemplars by performing belief propagation in a graphical model. (b) Two kinds of message are passed through in the graph, which are calculated based on the input similarity. Every data point will receive availability $a(i,k)$ and send responsibility $r(i,j)$ from/to node k, where $a(i,k)$ indicates how well the node k as exemplar for i and $r(i,k)$ indicates the possibility that the node k is chosen as an exemplar for i.

Algorithm 1 Affinity Propagation (AP) ([11]).

Input: A set of data points $\{x_i\}_{i=1}^n \in \mathbb{R}^{m\times n}$, and the similarity between pairs of data points $s(i,k)$.

1: **Initialization**:
$r(i,k) = 0$, $a(k,i) = 0$ for all i, k.

2: **Responsibility updates**:
$r(i,k) \longleftarrow s(i,k) - \max_{j:j\neq k}(a(j,i) + s(i,j))$.

3: **Availability updates**:
$a(k,k) \longleftarrow \max_{j:j\neq k}(a(j,i) + s(i,j))$
$a(k,i) \longleftarrow \min(0, r(k,k) + \sum_{j:j\notin\{k,i\}} \max(0, r(j,k)))$

4: **Making assignments**:
$c_i^* \longleftarrow \operatorname{argmax}_k\ r(i,k) + a(k,i)$.

Output: Cluster assignments $\{c_i^*\}_{i=1}^n$, where $c_i^* = i$ indicates the ith data point is chosen as an exemplar.

candidate exemplars. Message $r(i,k)$ is sent from point i to candidate exemplar k, which denotes how well point k is selected as exemplar for i. Moreover, $a(i,k)$ is sent from exemplar k to data point i, which indicates the possibility for point i would choose k as its exemplar. Every data point will be cluster member and candidate exemplar, simultaneously. The role of point only depends on which kind of message is sent. The update of responsibility leads to the competition in all candidate exemplars for the ownership of a data point, and the update of availability shows whether the corresponding exemplar would be a good exemplar. In other words, $a(k,k) = P(c_k = k)$ denotes the probability for point k would be chosen as an exemplar, and $r(i,k) = P(x_i|x_k)$ denotes the probability for point i belonging to the cluster k.

The performance of the AP algorithm is sensitive to the initialization, i.e., the similarity graph lies in the heart of the algorithm. In general, AP obtains a similarity graph using the negative Euclidean distance between any two data points, i.e., $s(i,j) = -\|x_i - x_j\|_2$. However, the distance metric would degrade the performance of AP in high-dimensional data analysis since the data locate in the

non-Euclidean space. Thus, it is necessary to find a more suitable measurement for the AP algorithm.

III. Sparse Affinity Propagation

Sparse representation assumes that each data point $x \in \mathbb{R}^m$ can be encoded as a linear combination of other points. In mathematically, $x = Dc$, where D is a dictionary whose columns consist of some data points, and c is the representation of x over D. If most entries of c are zeros, then c is called a sparse representation. Generally, it can be achieved by solving

$$\min \|c\|_0 \quad \text{s.t.} \quad x = Dc, \tag{1}$$

where $\|\cdot\|_0$ denotes ℓ_0-norm by counting the number of nonzero entries in a vector.

However, ℓ_0-minimization problem is difficult to solve, for it is a NP-hard problem. Benefiting from the emergence of compressive sensing theory [19], [20], the above non-convex problems can be solved by minimizing a convex surrogate when c is highly sparse, i.e.,

$$\min \|c\|_1 \quad \text{s.t.} \quad x = Dc, \tag{2}$$

where ℓ_1-norm $\|\cdot\|_1$ sums the absolute value of all entries in a vector. (2) is convex and can be solved by a large amount of convex optimization methods such as basis pursuit [21], least angle regression [22], and augmented lagrange multiplier method [23]. In [24], Yang et al. give a comprehensive survey for some popular optimizers.

In many practical problems, however, the constraint $x = Dc$ cannot hold exactly owing to the possible noises in x and the limited representational capability of D. So, we relax the constraint to $\|x - Dc\|_2 \leq \varepsilon$, where $\varepsilon > 0$ is the error tolerance. Then, (2) is rewritten as:

$$\min \|c\|_1 \quad \text{s.t.} \quad \|x - Dc\|_2 \leq \varepsilon. \tag{3}$$

For a given data set, we aims to find a small portion of data set (i.e., exemplars) that can represent the whole data set. Thus, we use all data points except x_i as dictionary for x_i, i.e.,

$$\min \|c_i\|_1 \quad \text{s.t.} \quad \|x_i - X_i c_i\|_2 \leq \varepsilon, \tag{4}$$

where c_i is the sparse representation of the data point x_i over $X_i := [x_1, x_2, \cdots, x_{i-1}, 0, x_{i+1}, \cdots, x_n]$.

The ℓ_1 optimization program could choose the most representative data points to reconstruct each data point x_i. However, the sparsity caused by ℓ_1-minimization is one-dimensional, i.e., the ℓ_1-norm is enforced over a vector such that it only guarantees to choose the minimal data points to represent each point but the whole data set. Thus, the results of (4) cannot be directly used for finding exemplars. Fig. 2 gives a toy example to show the situation. We perform (4) over a subset of AR database [25], where Fig. 2(a) shows some sample images. From Fig. 2(b), we could see that the affinity matrix corresponding to the collection of c_i is a block-diagonal matrix. For each data point x, only a small number of data points are used to represent x, but all data points are used to represent the whole data set (see Fig. 2(c)). This corroborates the above claims.

Algorithm 2 Sparse Affinity Propagation (SAP).

Input: A set of data points $\{x_i\}_{i=1}^n \in \mathbb{R}^{m\times n}$, and error tolerance ε.

1: Calculate the sparse representation for each data point by solving

$$\min \|c_i\|_1 \quad \text{s.t.} \quad \|x_i - X_i c_i\|_2 \leq \varepsilon.$$

2: Build an affinity matrix via

$$s(i,j) = |c(i,j)| + |c(j,i)|,$$

where $c(i,j)$ is the jth entry in c_i.

3: Initialization:
$r(i,k) = 0$, $a(k,i) = 0$ for all i, k.

4: Responsibility updates:
$r(i,k) \longleftarrow s(i,k) - \max_{j:j\neq k}(a(j,i) + s(i,j))$.

5: Availability updates:
$a(k,k) \longleftarrow \max_{j:j\neq k}(a(j,i) + s(i,j))$
$a(k,i) \longleftarrow \min(0, r(k,k) + \sum_{j:j\notin\{k,i\}} \max(0, r(j,k)))$

6: Making assignments:
$c_i^* \longleftarrow \operatorname{argmax}_k\ r(i,k) + a(k,i)$.

Output: Cluster assignments $\{c_i^*\}_{i=1}^n$, where $c_i^* = i$ indicates the ith data point is chosen as an exemplar.

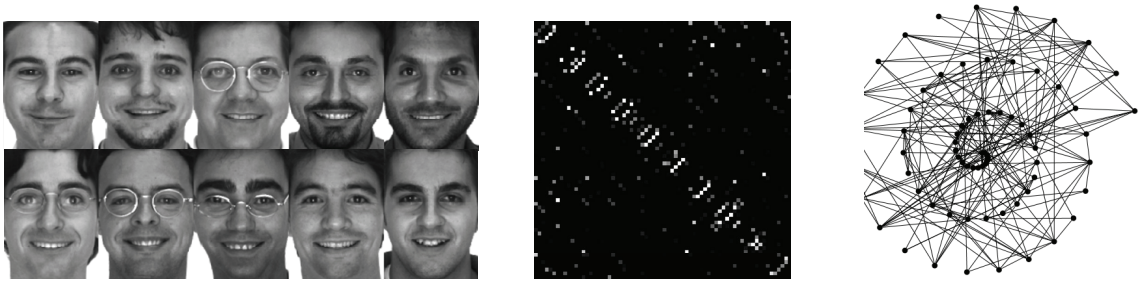
To find the minimal data points to represent the whole data set, we propose Sparse Affinity Propagation (SAP) by performing AP over the collection of the sparse representation. From the view of sparse coding, SAP adopts AP to eliminate the redundancy of sparse representation caused by transmissibility of linear encoding. In other words, SAP provides 2-dimensional sparsity by introducing the competition in all data points for the ownership of a data point. From the view of the AP algorithm, SAP takes the sparse representation as input to calculate the belief messages such that the algorithm is more competitive in high-dimensional data analysis. Algorithm 2 summarizes our algorithm, where Steps 3 - 6 is to perform AP over the sparse similarity graph.

IV. Experimental Verification and Analysis

In this section, we report the performance of SAP in the context of image classfication from accuracy, robustness, and computational cost. All experiments are carried out over a E5200 2.5Ghz CPU and 2GB memory in MATLAB.

A. Experimental configurations

We compared SAP with three popular methods, i.e., AP [11], kmedoids [1], and random sampling. SAP and AP automatically identify the exemplars without a fixed parameter, while kmedoids and random sampling need to specify the number of desired exemplars. For each data set, we perform SAP, AP, kmedoids and random sampling

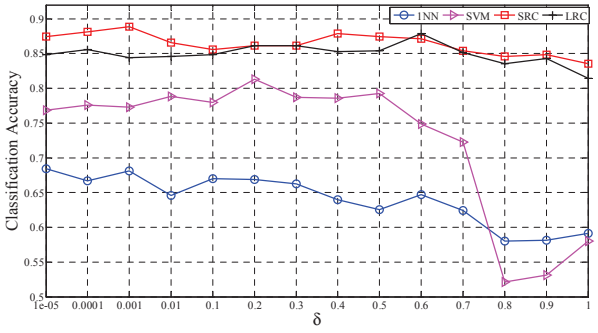


(a) (b) (c)

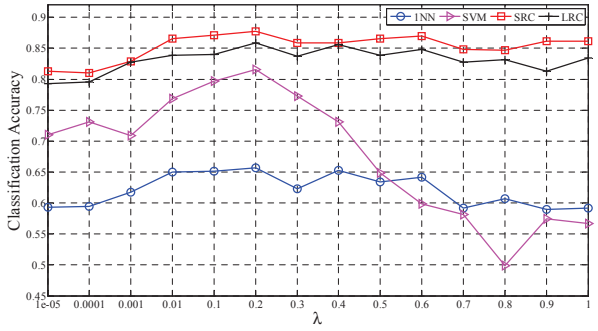
Figure 2. A toy example. (a) Some samples draw from the first 10 individuals of AR database. (b) The affinity matrix based on sparse representation. (c) Similarity graph based on sparse representation.

over training data to find exemplars, and then use the exemplars as training data to classify new data. Four state-of-the-art classifiers are used, i.e., SVM with linear kernel [26], 1NN classifier [27], Sparse Representation based Classifier (SRC) [28], and Linear Regression Classifier (LRC) [29]. In our experiments, SAP and SRC adopt Homotopy optimizer [30] to solve the ℓ_1-minimization problem (4). Similar to [31], in all experiments, we tuned the parameters of all methods to get their best results.

We examined the performance of the algorithms using two facial image databases and one handwritten digital database, i.e., Extended Yale Database B (YaleB) [32], AR [25], and USPS [27]. The YaleB data set contains 2414 frontal-face images with size 192×168 over 38 subjects. The AR data set includes over 4000 face images of 126 people where each subject has 26 images consisting of 14 clean images, 6 images with sunglasses and 6 images with scarves. As did in [33], a subset that contains 1400 clean faces randomly selected from 50 male subjects and 50 female subjects, is used in our experiment. USPS is a widely-used digital data set, which consists of 11000 handwritten digital images with 256 dimensionality over 10 classes. For all the databases, we randomly split the data into two parts of equal size, one for training, and the other for testing. Moreover, we cropped and normalized YaleB images from 192×168 to 54×48 and AR images from 165×120 to 55×40. For computational efficiency, we perform Principal Component Analysis (PCA) [34], [35] to reduce the dimensionality of the data.



(a)



(b)

Figure 3. Recognition accuracy of SAP with four classifiers over AR data set, where 167 features are extracted by PCA. (a) the recognition rates versus the varying δ, where $\lambda = 0.1$. (b) the recognition rates versus the varying λ, where $\delta = 0.1$.

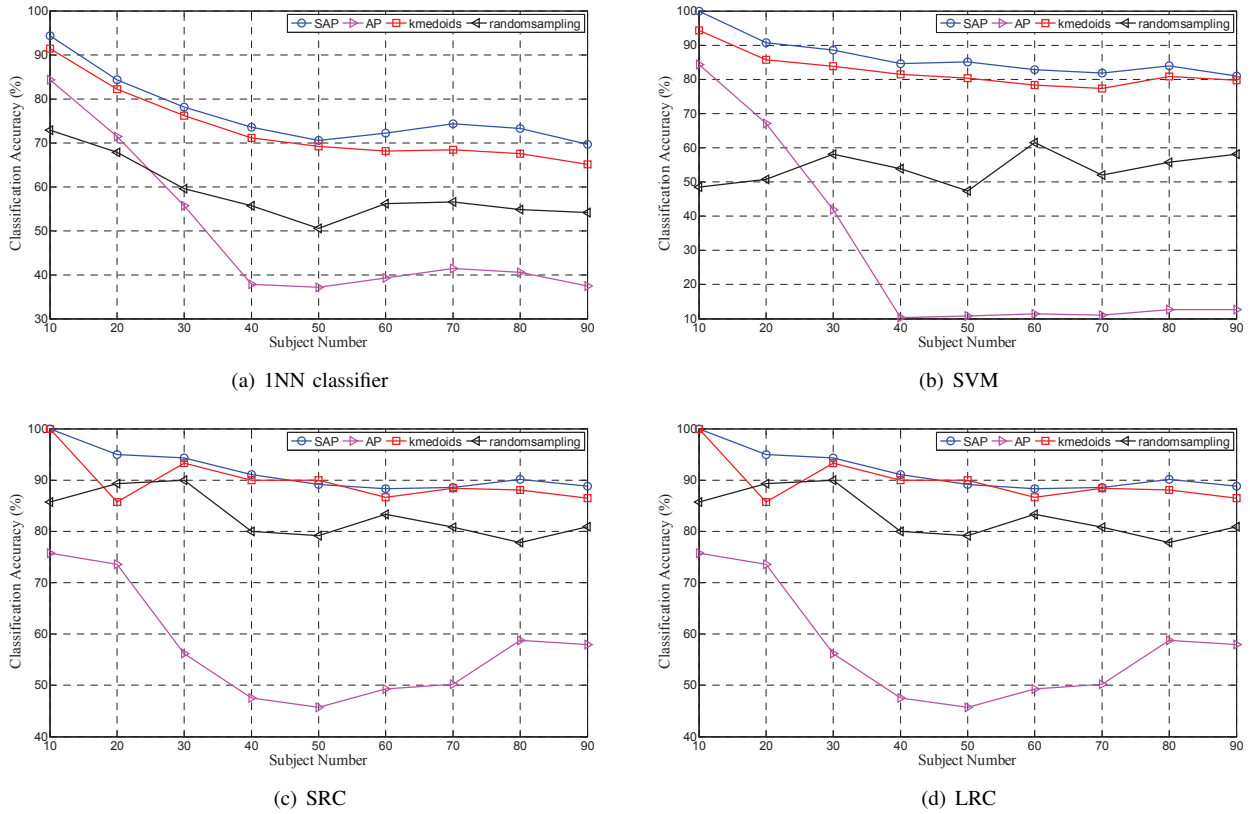
B. Model Selection

It is challenging to determine the value of parameters in pattern recognition and computer vision. In this subsection, we report the performance of SAP with the different value for the required parameters. SAP only needs to pre-specify the parameters for solving ℓ_1-minimization problem (4). In our experiments, Homotopy optimizer, which is adopted to achieve the sparse representation, requires pre-determining two parameters, i.e., sparsity parameter λ and error tolerance parameter δ.

We carried out experiments using a subset of AR database which consists of 1400 clean images over 100 individuals. Fig. 3 shows the classification results of SAP with 1NN classifier, SVM, SRC, and LRC. From the results, we could see that SRC and LRC perform more stable than 1NN and SVM. Moreover, the classification accuracies of the 1NN classifier and SVM decrease as the parameters increase. For all the used classifiers, SRC and LRC obtain similar recognition rate, which are superior to SVM and 1NN by considerable performance margins.

C. Classification on Clean Images

In this subsection, we investigate the performance of SAP with varying subject number using clean AR images. Moreover, we carry out the experiments using AR database with 100 subjects, YaleB with 38 subjects, and USPS with 10 subjects. In the experiments, SAP and AP automatically determinate the number of exemplars,



(a) 1NN classifier (b) SVM

(c) SRC (d) LRC

Figure 4. Classification results of SAP, AP, kmedoids, and random sampling with four different classifiers. The experiments are carried out over the images belonging to first k subjects of AR database, where k increases from 10 to 90 with an interval of 10.

while kmedoids and random sampling set the number of exemplar as triple of the number of subject.

Fig. 4 shows the classification results of four different classifiers with SAP, AP, kmedoids, and random sampling. From the results, we make the following observations:

- SAP generally outperforms the other models under the same experimental configuration. For example, the accuracy of SAP is at least 15.71% higher than AP, 1.21% higher than kmedoids, and 21.43% higher than random sampling, when SVM is adopted.
- SAP is more robust than AP which adopts negative Euclidean distance to measure the similarity among data points. Moreover, AP generally achieves the worst performance in the tests. This verifies the claim that SAP is more competitive than AP in high-dimensional data analysis.
- SRC and LRC perform very similar and are superior to 1NN and SVM. Moreover, SVM is more competitive than 1NN.
- With the increase in the number of subject (k), the recognition rate of AP has fallen by half. Especially, AP is failed to find a good set of exemplars when $k > 40$.

Table I - III report the results of SAP with four different classifiers using AR, YaleB, and USPS data set, from which we make the following observations:

- The SAP algorithm outperforms the other methods in the tests, whereas AP achieves the worst results. For example, SAP is at least 30.57%, 48.09%, 54.96% higher than AP using AR, YaleB, and USPS, respectively.
- For facial database (AR and YaleB), SAP identifies less exemplars to represent the whole data set, comparing with kmedoids and random sampling. For example, SAP finds about 280 and 350 exemplars for the AR and the YaleB, and the corresponding numbers for kmedoids or random sampling are 300 and 380.
- SAP demonstrates an elegant balance between running time and classification quality. Although SAP requires more time to find the representative data points, it achieves an obvious improvement in recognition rates. For example, SAP is at least 8.28% higher than the second best method using USPS.

D. Classification on Corrupted Images

It is inevitable in real applications that the data are contaminated by various corruptions. In this subsection, we investigate the robustness of SAP to two kinds of corruption using the AR data set. For each subject, we randomly choose half of the images (7 samples per subject) and corrupt them using white Gaussian noise or random pixel corruption as [31] did. Then, we divide the 14 images into two groups of equal size, one for training and the other for testing. Thus, both the training data and the testing data may be contaminated by noise. Fig. 5 shows some samples with different levels of corruption.

TABLE I.
RECOGNITION RATE AND TIME COST OF FOUR EXEMPLARS IDENTIFYING METHODS USING THE AR DATA SET (100 SUBJECTS). EACH AR IMAGE IS REDUCED TO 167 DIMENSION BY USING PCA. NUMBERS IN PARENTHESES INDICATE TUNED PARAMETERS FOR ACHIEVING THE BEST ACCURACY. SPECIFICALLY, SAP(λ, δ, NUMBER OF EXEMPLARS), AP (THE SIZE OF NEIGHBORHOOD, NUMBER OF EXEMPLARS), KMEDOIDS (NUMBER OF EXEMPLARS), AND RANDOM SAMPLING (NUMBER OF EXEMPLARS).

Classifiers	SAP		AP		kmedoids		random sampling	
	accuracy	time	accuracy	time	accuracy	time	accuracy	time
1NN	**66.71** (0.1, 0.1, 281)	4.61	36.14 (1, 103)	1.80	63.71 (300)	4.38	51.29 (300)	0.07
SVM	**80.86** (0.01, 0.2, 281)	4.20	12.43 (1, 103)	1.96	79.00 (300)	4.81	52.57 (300)	0.72
SRC	**88.29** (0.1, 1e-3, 274)	24.07	55.86 (1, 103)	16.29	85.57 (300)	25.08	80.43 (300)	21.22
LRC	86.14 (0.1, 0.1, 281)	5.79	45.29 (1, 103)	2.85	**91.71** (300)	5.89	84.29 (300)	1.22

TABLE II.
RECOGNITION RATE AND TIME COST OF FOUR EXEMPLARS IDENTIFYING METHODS USING THE EXTENDED YALE DATABASE B (YALEB WITH 38 SUBJECTS). EACH IMAGE IS REDUCED TO 167 DIMENSION BY USING PCA. NUMBERS IN PARENTHESES INDICATE TUNED PARAMETERS FOR ACHIEVING THE BEST ACCURACY. SPECIFICALLY, SAP(λ, δ, NUMBER OF EXEMPLARS), AP (THE SIZE OF NEIGHBORHOOD, NUMBER OF EXEMPLARS), KMEDOIDS (NUMBER OF EXEMPLARS), AND RANDOM SAMPLING (NUMBER OF EXEMPLARS).

Classifiers	SAP		AP		kmedoids		random sampling	
	accuracy	time	accuracy	time	accuracy	time	accuracy	time
1NN	**65.97** (0.1, 1e-3, 351)	7.07	17.88 (1, 107)	2.58	48.73 (380)	5.49	60.07 (380)	0.05
SVM	**88.48** (1e-3, 0.1, 358)	7.07	21.69 (1, 107)	2.67	59.53 (380)	5.55	75.68 (380)	0.15
SRC	**90.65** (0.01, 0.01, 343)	42.12	24.05 (1, 107)	27.05	69.51 (380)	28.71	84.76 (380)	33.06
LRC	**94.19** (1e-4, 0.1, 355)	7.73	10.07 (1, 107)	3.08	41.38 (380)	5.96	89.56 (380)	0.83

TABLE III.
RECOGNITION RATE AND TIME COST OF FOUR EXEMPLARS IDENTIFYING METHODS USING THE USPS DATA SET WITH 256 DIMENSIONALITY. NUMBERS IN PARENTHESES INDICATE TUNED PARAMETERS FOR ACHIEVING THE BEST ACCURACY. SPECIFICALLY, SAP(λ, δ, NUMBER OF EXEMPLARS), AP (THE SIZE OF NEIGHBORHOOD, NUMBER OF EXEMPLARS), KMEDOIDS (NUMBER OF EXEMPLARS), AND RANDOM SAMPLING (NUMBER OF EXEMPLARS).

Classifiers	SAP		AP		kmedoids		random sampling	
	accuracy	time	accuracy	time	accuracy	time	accuracy	time
1NN	**94.36** (0.1, 0.1, 1624)	162.32	39.40 (1, 52)	32.13	85.24 (100)	68.45	67.75 (100)	0.13
SVM	**91.64** (0.01, 0.1, 1626)	142.17	32.40 (1, 52)	32.21	83.36 (100)	68.53	71.38 (100)	0.19
SRC	**96.64** (0.01, 1e-3, 1511)	702.55	40.51 (1, 52)	127.54	86.06 (100)	176.78	72.49 (100)	128.19
LRC	**90.22** (1e-4, 0.2, 1636)	169.70	31.66 (1, 52)	32.96	74.06 (100)	69.43	66.98 (100)	1.47

TABLE IV.
RECOGNITION RATE AND TIME COST OF FOUR EXEMPLARS IDENTIFYING METHODS WITH THE 1NN CLASSIFIER USING CORRUPTED AR IMAGES. NUMBERS IN PARENTHESES INDICATE TUNED PARAMETERS FOR ACHIEVING THE BEST ACCURACY. SPECIFICALLY, SAP(λ, δ, NUMBER OF EXEMPLARS), AP (THE SIZE OF NEIGHBORHOOD, NUMBER OF EXEMPLARS), KMEDOIDS (NUMBER OF EXEMPLARS), AND RANDOM SAMPLING (NUMBER OF EXEMPLARS).

Corruptions	Corruption Rate	SAP		AP		kmedoids		random sampling	
		accuracy	time	accuracy	time	accuracy	time	accuracy	time
Gaussian	10	**62.00** (0.001,0.01,20)	5.88	54.71 (1,200)	2.13	59.71 (200)	4.02	51.14 (200)	0.05
	30	**47.00** (0.1,0.2,208)	3.51	27.00 (1,103)	1.81	44.71 (200)	3.77	33.57 (200)	0.05
	50	**54.57** (0.001,0.2,216)	3.70	53.14 (1,199)	2.21	45.14 (200)	4.11	31.43 (200)	0.04
	70	**64.29** (0.01,0.001,20)	3.90	57.71 (1,200)	2.03	34.00 (200)	3.73	35.71 (200)	4.23
	90	**43.86** (0.1,0.001,287)	3.68	11.29 (1,102)	1.81	32.29 (200)	3.92	26.71 (200)	0.04
Random Pixel Corruption	10	**52.14** (0.01,0.2,299)	3.69	49.29 (1,105)	1.78	39.00 (200)	29.74	32.14 (200)	0.04
	30	**49.57** (1e-5,0.2,207)	3.37	35.14 (1,118)	1.99	38.14 (200)	21.30	20.86 (200)	0.04
	50	**23.71** (0.001,0.2,247)	3.15	12.43 (1,101)	1.47	22.43 (200)	21.91	15.00 (200)	0.05
	70	**27.43** (0.001,0.2,276)	3.38	5.57 (1,100)	1.43	15.14 (200)	23.22	17.57 (200)	0.04
	90	27.40 (1e-5,0.2,201)	2.97	2.43 (1,100)	1.41	**27.86** (200)	3.67	16.86 (200)	0.03

Table IV to Table VII report the performance of SAP with 1NN, SVM, SRC, and LRC classifiers, from which we can see that

- SAP is more robust to the two kinds of corruptions than AP, kmedoids, and random sampling by considerable performance margins. For example, when data contain white Gaussian noise, the accuracy of SAP with the 1NN classifier over that of AP in the range of [+1.43% +32.57%]. The corresponding numbers over kmedoids and random sampling are [+2.29% +30.29%] and [+13.43% +28.58%], respectively.
- SAP with SRC achieves the best result in the tests. Their recognition rate is about 58.29% even though 90% data are contaminated by Gaussian noise, whereas the second best combination (kmedoids with SRC) only obtains a 40.43% accuracy.

TABLE V.
RECOGNITION RATE AND TIME COST OF FOUR EXEMPLARS IDENTIFYING METHODS WITH THE SVM CLASSIFIER USING CORRUPTED AR IMAGES. NUMBERS IN PARENTHESES INDICATE TUNED PARAMETERS FOR ACHIEVING THE BEST ACCURACY. SPECIFICALLY, SAP(λ, δ, NUMBER OF EXEMPLARS), AP (THE SIZE OF NEIGHBORHOOD, NUMBER OF EXEMPLARS), KMEDOIDS (NUMBER OF EXEMPLARS), AND RANDOM SAMPLING (NUMBER OF EXEMPLARS).

Corruptions	Corruption Rate	SAP		AP		kmedoids		random sampling	
		accuracy	time	accuracy	time	accuracy	time	accuracy	time
Gaussian	10	**63.43** (1e-5,0.01,201)	6.91	45.29 (1,200)	2.34	62.29 (200)	4.23	52.14 (200)	0.30
	30	46.00 (0.01,0.1,207)	4.49	21.29 (1,103)	1.97	**46.57** (200)	3.97	40.57 (200)	0.31
	50	**62.14** (0.1,0.001,212)	4.35	54.14 (1,199)	2.42	51.86 (200)	4.32	30.43 (200)	0.23
	70	**56.86** (0.1,0.001,203)	3.54	59.43 (1,200)	2.23	47.00 (200)	3.94	31.86 (200)	23.04
	90	**55.86** (0.01,0.1,279)	4.58	12.57 (1,102)	1.98	40.29 (200)	4.10	22.71 (200)	0.23
Random Pixel Corruption	10	**78.86** (0.1,0.001,295)	4.12	9.14 (1,105)	1.93	70.71 (200)	29.91	33.71 (200)	0.22
	30	**46.43** (0.1,0.2,206)	3.85	23.86 (1,118)	2.16	34.57 (200)	21.48	29.43 (200)	0.23
	50	**30.29** (0.01,0.1,242)	3.75	10.86 (1,101)	1.99	25.43 (200)	22.09	16.29 (200)	0.23
	70	**28.71** (1e-5,0.2,276)	3.63	10.71 (1,100)	1.54	16.00 (200)	23.39	15.57 (200)	0.23
	90	**18.00** (0.01,0.01,224)	5.21	3.29 (1,100)	1.53	17.86 (200)	3.89	9.43 (200)	0.24

TABLE VI.
RECOGNITION RATE AND TIME COST OF FOUR EXEMPLARS IDENTIFYING METHODS WITH THE SRC CLASSIFIER USING CORRUPTED AR IMAGES. NUMBERS IN PARENTHESES INDICATE TUNED PARAMETERS FOR ACHIEVING THE BEST ACCURACY. SPECIFICALLY, SAP(λ, δ, NUMBER OF EXEMPLARS), AP (THE SIZE OF NEIGHBORHOOD, NUMBER OF EXEMPLARS), KMEDOIDS (NUMBER OF EXEMPLARS), AND RANDOM SAMPLING (NUMBER OF EXEMPLARS).

Corruptions	Corruption Rate	SAP		AP		kmedoids		random sampling	
		accuracy	time	accuracy	time	accuracy	time	accuracy	time
Gaussian	10	**75.86** (0.1,0.001,213)	23.51	69.86 (1,200)	20.24	73.57 (200)	21.70	72.14 (200)	24.81
	30	60.14 (1e-5,0.2,208)	23.36	28.86 (1,103)	19.46	**60.29** (200)	25.44	54.57 (200)	31.68
	50	**78.57** (0.01,0.1,216)	23.79	58.57 (1,199)	21.91	65.00 (200)	22.30	49.00 (200)	24.91
	70	**70.71** (0.001,0.01,205)	23.78	64.14 (1,200)	19.28	44.14 (200)	21.72	48.71 (200)	24.40
	90	**58.29** (0.01,0.1,279)	24.88	21.14 (1,102)	22.51	40.43 (200)	19.69	38.43 (200)	21.89
Random Pixel Corruption	10	**82.57** (0.1,0.001,295)	27.78	44.43 (1,105)	20.40	76.86 (200)	42.57	56.43 (200)	24.04
	30	**66.57** (0.001,0.001,204)	25.83	44.14 (1,118)	20.72	55.14 (200)	35.10	41.86 (200)	25.98
	50	**48.86** (0.001,0.1,240)	23.56	19.14 (1,101)	14.44	40.86 (200)	35.49	28.00 (200)	27.36
	70	**32.86** (1e-5,0.2,276)	22.41	9.29 (1,100)	14.63	19.57 (200)	35.90	24.57 (200)	26.10
	90	30.71 (0.01,0.2,201)	23.41	3.00 (1,100)	13.61	**31.00** (200)	26.02	19.43 (200)	26.49

TABLE VII.
RECOGNITION RATE AND TIME COST OF FOUR EXEMPLARS IDENTIFYING METHODS WITH THE LRC CLASSIFIER USING CORRUPTED AR IMAGES. NUMBERS IN PARENTHESES INDICATE TUNED PARAMETERS FOR ACHIEVING THE BEST ACCURACY. SPECIFICALLY, SAP(λ, δ, NUMBER OF EXEMPLARS), AP (THE SIZE OF NEIGHBORHOOD, NUMBER OF EXEMPLARS), KMEDOIDS (NUMBER OF EXEMPLARS), AND RANDOM SAMPLING (NUMBER OF EXEMPLARS).

Corruptions	Corruption Rate	SAP		AP		kmedoids		random sampling	
		accuracy	time	accuracy	time	accuracy	time	accuracy	time
Gaussian	10	68.29 (1e-5,0.2,216)	4.99	46.86 (1,200)	3.09	60.57 (200)	5.04	**79.71** (200)	1.19
	30	39.57 (0.1,0.01,205)	4.71	12.71 (1,103)	3.06	32.29 (200)	4.68	**60.43** (200)	1.20
	50	**44.57** (0.001,0.2,216)	4.65	35.71 (1,199)	3.19	40.57 (200)	5.04	33.14 (200)	1.09
	70	**33.43** (0.01,0.001,204)	4.84	30.29 (1,200)	3.07	30.00 (200)	4.64	27.29 (200)	1.08
	90	**42.43** (0.1,0.001,287)	4.79	11.86 (1,102)	3.02	30.14 (200)	4.85	20.57 (200)	1.08
Random Pixel Corruption	10	**76.86** (0.01,0.1,298)	4.95	36.29 (1,105)	2.94	42.57 (200)	30.56	36.00 (200)	1.05
	30	**35.57** (0.001,0.01,207)	0.03	18.71 (1,118)	3.18	29.00 (200)	22.13	24.71 (200)	1.09
	50	**42.14** (0.0001,0.2,247)	4.08	7.00 (1,101)	2.33	27.43 (200)	22.74	21.14 (200)	1.08
	70	**17.43** (0.001,0.2,276)	4.56	6.29 (1,100)	2.27	9.00 (200)	24.03	11.43 (200)	1.06
	90	**25.57** (0.001,0.01,225)	5.40	2.71 (1,100)	2.26	22.57 (200)	4.64	14.86 (200)	1.10

- The tested models perform worse as increasing corruption ratio. Moreover, random pixel corruption is more challenging than white Gaussian noise, where the former is a kind of non-additive noise and the latter is additive.

E. Recognition with Real Occlusions

To investigate the robustness of the algorithms to real occlusions, we carry out the experiments using two subsets of the AR database. The first subset consists of 600 clean images and 600 faces with sunglasses over 100 individuals, and the second subset consists of 600 clean images and 600 faces with scarves. Fig. 6 shows some sample images.

Table VIII through Table IX are the recognition rates and time costs of the tested methods with four classifiers. Clearly, SAP again outperforms the other approaches in the experiments. When the faces are occluded by sunglasses or scarves, the accuracy of SAP with LRC is 11.17% and 12.50% at least higher than the second best method (kmedoids with LRC). For the used classifiers,

TABLE VIII.
RECOGNITION RATE AND TIME COST OF FOUR EXEMPLARS IDENTIFYING METHODS USING THE AR IMAGES WITH SUNGLASSES (143D). NUMBERS IN PARENTHESES INDICATE TUNED PARAMETERS FOR ACHIEVING THE BEST ACCURACY. SPECIFICALLY, SAP(λ, δ, NUMBER OF EXEMPLARS), AP (THE SIZE OF NEIGHBORHOOD, NUMBER OF EXEMPLARS), KMEDOIDS (NUMBER OF EXEMPLARS), AND RANDOM SAMPLING (NUMBER OF EXEMPLARS).

Classifiers	SAP		AP		kmedoids		random sampling	
	accuracy	time	accuracy	time	accuracy	time	accuracy	time
1NN	21.17 (0.01,0.2,234)	3.21	16.67 (1,145)	1.96	20.00 (250)	45.51	**23.67** (250)	0.03
SVM	**50.00** (0.1,0.1,235)	3.41	29.83 (1,145)	2.12	48.67 (250)	45.42	35.67 (250)	0.22
SRC	**58.33** (0.1,0.2,234)	24.11	41.00 (1,145)	20.95	55.83 (250)	60.25	50.17 (250)	15.98
LRC	**78.00** (0.1,0.2,234)	3.86	12.83 (1,145)	2.86	66.83 (250)	46.09	66.33 (250)	0.90

TABLE IX.
RECOGNITION RATE AND TIME COST OF FOUR EXEMPLARS IDENTIFYING METHODS USING THE AR IMAGES WITH SCARVES (147D). NUMBERS IN PARENTHESES INDICATE TUNED PARAMETERS FOR ACHIEVING THE BEST ACCURACY. SPECIFICALLY, SAP(λ, δ, NUMBER OF EXEMPLARS), AP (THE SIZE OF NEIGHBORHOOD, NUMBER OF EXEMPLARS), KMEDOIDS (NUMBER OF EXEMPLARS), AND RANDOM SAMPLING (NUMBER OF EXEMPLARS).

Classifiers	SAP		AP		kmedoids		random sampling	
	accuracy	time	accuracy	time	accuracy	time	accuracy	time
1NN	**21.83** (0.1,0.1,241)	3.22	16.00 (1,148)	2.14	20.67 (250)	50.03	20.83 (250)	2.84
SVM	41.83 (0.1,0.2,246)	3.27	30.00 (1,148)	2.39	**45.50** (250)	50.21	32.17 (250)	0.24
SRC	**55.67** (0.1,0.1,241)	25.14	41.50 (1,148)	21.20	50.83 (250)	65.19	44.67 (250)	15.88
LRC	**73.00** (0.1,0.01,234)	4.34	19.67 (1,148)	3.09	60.50 (250)	50.78	60.00 (250)	0.91

Figure 5. Some sample images with corruptions. From left to right, the corruption ratio is 10, 30, 50, 70, and 90%. The images in the first row are contaminated by Gaussian noise; images in the second row are contaminated by random pixel corruption.

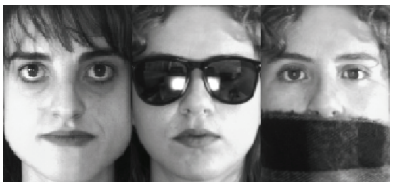


Figure 6. Some sample images with real disguises, where the occlusion rate of sunglasses is about 20% and that of scarves is about 40%.

LRC is superior to 1NN, SVM, and SRC. Moreover, the methods achieves similar recognition rate even though the occlusion ratio is largely different in two disguises.

V. CONCLUSIONS

In this paper, we proposed using sparse representation to construct a similarity graph for the AP algorithm, The proposed algorithm, which is named as Sparse Affinity Propagation (SAP), eliminates the redundancy of sparse representation caused by transmissibility of linear encoding such that a small set of data points could be identified to reconstruct the whole data set. In other words, SAP achieves the two-dimensional sparsity by introducing the competition in all data points for the ownership of a data point. From the other hand, SAP takes the sparse representation as input to calculate the belief messages such that the algorithm is more competitive in high-dimensional data analysis. Extensive experimental studies shows SAP outperforms AP, kmedoids, and random sampling in the context of image classification.

Each approach has its own advantages and disadvantages. The improvement in accuracy of SAP is at cost of computation time. SAP requires to solve a ℓ_1-minimization problem whose time complexity is proportion to the cube of data size, i.e., $O(n^3)$ versus $O(n^2)$ of the standard AP algorithm. Therefore, it is interesting to explore the way to reduce the time cost.

REFERENCES

[1] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for k-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009.

[2] C. Boutsidis, M. W. Mahoney, and P. Drineas, "An improved approximation algorithm for the column subset selection problem," in *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2009, pp. 968–977.

[3] J. A. Tropp, "Column subset selection, matrix factorization, and eigenvalue optimization," in *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2009, pp. 978–986.

[4] C. Yang, J. Shen, J. Peng, and J. Fan, "Image collection summarization via dictionary learning for sparse representation," *Pattern Recognition*, 2012.

[5] E. Elhamifar, G. Sapiro, and R. Vidal, "See all by looking at a few: Sparse modeling for finding representative objects," in *Proceedings of IEEE Conference on Computer*

Vision and Pattern Recognition. IEEE, 2012, pp. 1600–1607.

[6] C. Yang, J. Shen, J. Peng, and J. Fan, "Image collection summarization via dictionary learning for sparse representation," *Pattern Recognition*, 2012.

[7] X. Peng, L. Zhang, and Z. Yi, "Scalable sparse subspace clustering," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[8] L. Lang and X. Jing, "Application of non-negative sparse matrix factorization in occluded face recognition," *Journal of Computers*, vol. 6, no. 12, pp. 2675–2679, 2011.

[9] X. Peng, L. Zhang, and Z. Yi, "Inductive sparse subspace clustering," *IET Electronics Letters (Accepted)*, 2013.

[10] B. Frey and D. Dueck, "Mixture modeling by affinity propagation," vol. 18, 2006, p. 379.

[11] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *science*, vol. 315, no. 5814, pp. 972–976, 2007.

[12] H. Zhu, S. Ding, X. Xu, and L. Xu, "A parallel attribute reduction algorithm based on affinity propagation clustering," *Journal of Computers*, vol. 8, no. 4, pp. 990–997, 2013.

[13] Y. Jia, J. Wang, C. Zhang, and X.-S. Hua, "Finding image exemplars using fast sparse affinity propagation," in *Proceedings of the ACM international conference on Multimedia*. ACM, 2008, pp. 639–642.

[14] I. Givoni, C. Chung, and B. J. Frey, "Hierarchical affinity propagation," *arXiv preprint arXiv:1202.3722*, 2012.

[15] Y. Wang and L. Gao, "Detecting protein complexes by an improved affinity propagation algorithm in protein-protein interaction networks," *Journal of Computers*, vol. 7, no. 7, pp. 1761–1768, 2012.

[16] D. Dueck, "Affinity propagation: clustering data by passing messages," Ph.D. dissertation, Citeseer, 2009.

[17] J. C. Lv, K. K. Tan, Z. Yi, and S. Huang, "Convergence analysis of a class of hyvarinencoja's ica learning algorithms with constant learning rates," *IEEE Transactions on Signal Processing*, vol. 57, no. 5, pp. 1811–1824, 2009.

[18] L. Zhen, X. Peng, and D. Peng, "Local neighborhood embedding for unsupervised nonlinear dimension reduction," *Journal of Software*, vol. 8, no. 2, pp. 410–417, 2013.

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

[20] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal ℓ^1-norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[21] S. S. B. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.

[22] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, no. 2, pp. 407–451, 2004.

[23] J. Yang and Y. Zhang, "Alternating direction algorithms for ℓ_1-problems in compressive sensing," *SIAM journal on scientific computing*, vol. 33, no. 1, pp. 250–278, 2011.

[24] A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Fast l1-minimization algorithms and an application in robust face recognition: A review," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2010-13, February 5 2010.

[25] A. Martinez and R. Benavente, "The ar face database," 1998.

[26] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, "Liblinear: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[27] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer Series in Statistics, 2001, vol. 1.

[28] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[29] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 2106–2112, 2010.

[30] M. R. Osborne, B. Presnell, and B. A. Turlach, "A new approach to variable selection in least squares problems," *IMA Journal of Numerical Analysis*, vol. 20, no. 3, pp. 389–403, 2000.

[31] X. Peng, L. Zhang, and Z. Yi, "Constructing l2-graph for subspace learning and segmentation," *arXiv:1209.0841*, 2012.

[32] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[33] X. Peng, L. Zhang, Z. Yi, and K. Kiong Tan, "Learning locality-constrained collaborative representation for face recognition," *arXiv:1210.1316*, 2012.

[34] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1991.

[35] J. C. Lv, K. K. Tan, Z. Yi, and S. Huang, "A family of fuzzy learning algorithms for robust principal component analysis neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 1, pp. 217–226, 2010.