Compressive Clustering of High-dimensional Data

Andrzej Ruta
Samsung Poland R&D Center
Warsaw, Poland
a.ruta@samsung.com

Fatih Porikli
Mitsubishi Electric Research Labs
Cambridge, MA, USA
fatih@merl.com

***Abstract*—In this paper we focus on realistic clustering problems where the input data is high-dimensional and the clusters have complex, multimodal distribution. In this challenging setting the conventional methods, such as k-centers family, hierarchical clustering or those based on model fitting, are inefficient and typically converge far from the globally optimal solution. As an alternative, we propose a novel unsupervised learning approach which is based on the compressive sensing paradigm. The key idea underlying our algorithm is to monitor the distance between the test sample and its principal projection in each cluster, and continue re-assigning it to the cluster yielding the smallest residual. As a result, we obtain an iterative procedure which, under the compressive assumptions, minimizes the total reconstruction error of all samples from their nearest clusters. To evaluate the proposed approach, we have conducted a series of experiments involving various image collections where the task was to automatically group similar objects. Comparison of the obtained results with those yielded by the state-of-the-art clustering methods provides evidence for high discriminative power of our algorithm.**

***Keywords*-compressive clustering; nearest subspace; residual minimization; coordinate descent**

I. Introduction

There exist many robust clustering algorithms that perform well on data exhibiting intrinsic grouping in the feature space. To this sort of data one can apply basic methods, such as connectivity-based algorithms [1], [2] or Lloyd's *k-means* [3], as long as cluster distributions are fairly regular and a simple analytical distance metric can be adopted. This idealized scenario is yet far from what we observe when tackling real-life problems: large number of groups, their irregular distributions and overlap, or presence of outliers. These factors usually make the above algorithms fail, especially for attribute-rich data, which renders particular distance functions indiscriminative in high-dimensional spaces.

To overcome early methods' limitations, different strategies for clustering were proposed. The most elegant approach treats clusters as grouping objects that belong most likely to the same distribution. This allows to produce complex statistical models of clusters and capture correlations of attributes in parallel. A whole family of generative methods emerged from this concept, including popular finite mixture models [4], [5]. They are usually trained using some variant of Expectation-Maximization algorithm (EM) [6] or Bayesian framework. The drawback of the above methods is that they put on the human the extra burden of choosing and optimizing a model which is inherently difficult as more complex models will usually explain the data better.

In density-based methods clusters are defined as areas of higher density than the remainder of the dataset. A good representative of this family is DBSCAN [7] and its numerous extensions. They are based on connecting points within certain distance thresholds as long as they satisfy some additional density criterion. The mean-shift clustering algorithm [8], also falling into this group, treats data points as drawn from an unknown distribution and attempts to iteratively find the nearest stationary point of the underlying density function. Density-based methods are generally robust, but typically require significant density drop to detect clusters and are hard to apply to high-dimensional data.

Another thread of research on clustering focused on tackling the curse of dimensionality which is a serious problem when attribute-rich data, e.g. multimedia, are involved. The corresponding methods attempt to identify only the relevant attributes to be included in the cluster models or look for arbitrarily rotated subspace clusters that can be modeled by giving a correlation of their attributes. Examples for such clustering algorithms are CLIQUE [9] and SUBCLU [10]. Automatic feature extraction methods for distribution-based clustering have also been proposed, e.g. in [11].

In recent years compressive sensing (CS) [12], [13] has attracted attention as a robust data compression method by which one can effectively represent in low-dimensional spaces high-dimensional signals sampled at sub-Nyquist frequencies. It holds on condition that the signal can be sparsely represented in a given basis. Wright et al. [14] employed CS to cope with pattern classification tasks by composing the sparsity basis of labeled training examples. If classes are sufficiently distinguishable, then the novel example will appear much closer to its linear projection onto the subspace spanned only by the training examples representing its true class than to its projection onto the subspace spanned by any other class. Chi and Porikli [15] extended the idea from [14] by combining the sparse representation based classifier with a classifier exploiting also the inter-class information encoded in the collaborative representation formed by all (as opposed to class-specific) training examples.

In this work we follow the intuition of Wright et al. in an unsupervised learning setting. The key idea of our novel algorithm is to operate on each high-dimensional data vector through its compressive, low-dimensional projection

onto a basis spanned by a subset of all other vectors from the dataset being explored. The nearest subspace for this vector should then correspond to the subset of points most similar to it. In other words, we rephrase the aforementioned assertion by stating that, under the compressive assumptions, a given vector should on average appear closer to its linear projection onto the subspace spanned by its true cluster than to its projection onto a subspace spanned by any other combination of vectors. This translates into an elegant residual minimization problem which we solve through an iterative coordinate descent algorithm [16]. This facilitates efficient data clustering which is demonstrated in the experiments involving various image collections.

The rest of this paper is composed as follows. In Section II the idea of compressive clustering, its theoretical foundations and implementation are discussed. In Section III we present a comparative evaluation of the proposed approach. Finally, in Section IV conclusions of the paper are drawn.

II. Compressive Clustering

In this chapter we first describe the way data points are sparsely represented. Then, the clustering task is posed as a residual minimization problem where the aim is to find the point-to-cluster assignment leading to possibly the smallest total error of reconstructing points from their clusters. Afterwards, an iterative algorithm solving the above problem is presented and its key implementation aspects are discussed.

A. Data Representation

Assume there are K distinct groups in the data containing N objects and further let $\mathbf{x}_i \in \mathbb{R}^M$ denote a feature vector describing the i-th object that needs to be assigned into one of these groups. Each such vector can be thought of as a superposition of all remaining vectors given a linear model:

$$\mathbf{x}_i = \mathbf{\Psi}_{\mathbf{x}_i}\alpha \ , \tag{1}$$

where $\mathbf{\Psi}_{\mathbf{x}_i}$ is an $M \times (N-1)$ matrix built from stacked vectors $\mathbf{x}_j$, $j \neq i$. We assume that vectors representing objects within the same group lie in the same low-dimensional linear subspace. Therefore, provided N is reasonably large, we expect each vector $\mathbf{x}_i$ to have sparse representation under its basis $\mathbf{\Psi}_{\mathbf{x}_i}$. In other words, we seek a sparse vector $\alpha = [\alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_N]$. It is done by solving an $\ell 1$-regularized least-squares regression problem [13]:

$$\alpha^* = \arg\min_{\alpha}\{\|\mathbf{\Phi}\mathbf{x}_i - \mathbf{\Phi}\mathbf{\Psi}_{\mathbf{x}_i}\alpha\|_{\ell 2}^2 + \lambda\|\alpha\|_{\ell 1}\} \ , \tag{2}$$

where $\mathbf{\Phi} \in \mathbb{R}^{D\times M}$ is a random Gaussian matrix, $D < M$, and λ is the regularization constant. D is chosen to satisfy:

$$D = \lceil \beta T \log N \rceil \ , \tag{3}$$

where T is the estimate of the number of non-zero weights in α and β is a constant set experimentally.

The matrix $\mathbf{\Phi}$ is used for dimensionality reduction. It is motivated by the compressive sensing theory [12] which states that the sparse pattern of a signal can be recovered from the heavily compressed, low-dimensional measurements, incoherent in the $\mathbf{\Psi}_{\mathbf{x}_i}$ domain. Although (if $M > N$) sparse α can be found by employing a conventional regularized least-squares approach, without signal compression, it often leads to overfitting due to noise and the insufficient number of measurements.

B. Residual Minimization Problem

With the above in mind, the goal of clustering is to find an optimal set of sparsity bases for each data point such that the sum of their reconstruction errors is minimized. For a given cluster membership function $c : \mathbb{R}^M \to \{1, \ldots, K\}$ and a given vector $\mathbf{x}_i$ the best cluster assignment update can be found by checking how well this vector gets reconstructed based on the found sparse vector α^* with set to zero all coefficients not corresponding to the basis vectors belonging to each j-th cluster. This way K signals are reconstructed:

$$\upsilon_j(\mathbf{x}_i) = \mathbf{\Psi}_{\mathbf{x}_i}\alpha_j^* \ , \tag{4}$$

where α_j^* denotes a vector of sparse coefficients with attenuated components outside the j-th cluster, and the cluster membership of $\mathbf{x}_i$ is found by minimizing the residual:

$$c^*(\mathbf{x}_i) = \arg\min_{j=1,\ldots,K} r_j(\mathbf{x}_i) = \arg\min_{j=1,\ldots,K}\{\|\mathbf{x}_i - \upsilon_j(\mathbf{x}_i)\|^2\}. \tag{5}$$

The result of the minimization in (5) depends on the current cluster contents, which can be quite random early in the process. Also, it only guarantees reduction of individual points' reconstruction errors, without ensuring global consistency of the resulting data partitioning. For effective search of the optimal solution to the problem, we formulate it as an optimization task with the following cost function:

$$E^* = \min_{\upsilon}\min_{c\in C}\sum_{j=1}^{K}\sum_{i:c(\mathbf{x}_i)=j}\|\mathbf{x}_i - \upsilon_j(\mathbf{x}_i)\|^2 \ , \tag{6}$$

where minimization over υ means searching the space of all possible zero weight assignments to the components of vector α^* for all data points and C is the space of all possible subdivisions of the input data into K clusters.

In Section II-C we present the algorithm used to solve the above problem and show its convergence. Its specific implementation issues are discussed in Section II-D.

C. Coordinate Descent Algorithm

To solve the residual minimization problem stated in the previous section, we propose an iterative procedure in principle similar to the popular Lloyd's algorithm, also known as Voronoi relaxation [3].

Note that the cost function in (6) depends on two parameters: υ and c. As the function itself is bounded, joint minimization over both parameters can be done by alternately fixing value of the first parameter and minimization over the second and fixing value of the second while minimizing the first. This scheme is known as *coordinate descent* and it is

proven to converge when the cost function is smooth [16]. Specifically, we first fix the point-specific subspaces encoded in υ and optimize c through linear search. We call it a *re-assignment* step. Subsequently, the newly found partitioning $\hat{c}$ is fixed, which yields:

$$\min_{\upsilon} \sum_{j=1}^{K} \sum_{i:\hat{c}(\mathbf{x}_i)=j} \|\mathbf{x}_i - \upsilon_j(\mathbf{x}_i)\|^2 = \sum_{j=1}^{K} \min_{\upsilon} \sum_{i:\hat{c}(\mathbf{x}_i)=j} \|\mathbf{x}_i - \upsilon_j(\mathbf{x}_i)\|^2 \quad . \tag{7}$$

Finding minima in the outer sum of (7) means applying equation (5) to all data points. As in each re-assignment step clusters become more consistent, in an ideal case of non-overlapping clusters the basis vectors that belong to one selected cluster gain predominant contribution to a given point's residual. Therefore, attenuating coefficients of basis vectors not belonging to this cluster is guaranteed to reduce (7), which was shown in [14] in the supervised learning context. However, noise and modeling errors will lead to small non-zero entries associated with multiple clusters which for difficult datasets may produce sub-optimal clustering results, just as for *k-means* and most other algorithms.

The proposed iterative optimization scheme can be summarized in the pseudocode shown in Algorithm 1. Assuming K is given, it starts by partitioning the input dataset into K subsets. Then, it iterates over all data vectors and for each it finds the cluster defining the nearest subspace according to (5) and computes this vector's contribution to the total residual. This is followed by re-assignment of the input vectors to their best found clusters which in turn triggers the new nearest subspace search. The algorithm is repeated by alternate application of these two steps until convergence.

Algorithm 1 Implementation of the proposed compressive clustering algorithm.

input: Dataset $\mathbf{X} = \{\mathbf{x}_i : \mathbf{x}_i \in \mathbb{R}^M\}$, K - the number of clusters
output: Optimal partitioning of data into clusters c^*

```
Initialize clusters c^(0) and calculate total residual E^(0)
Set E_min = E^(0), i = 1
while not converged do
    For each vector find its best cluster for re-assignment
    Re-assign vectors to their found best clusters, producing c^(i)
    Determine nearest subspaces and calculate total residual E^(i)
    if c^(i) ≡ c^(i-1) do Set convergence flag end if
    Set i = i + 1
end while
c* ← c^(i-1)
```

D. Implementation Issues

Several aspects of the proposed compressive clustering scheme require clarification. First, any cluster initialization method can be used, e.g. random one. Output of *k-means* or any other conventional algorithm is another possible option. However, care should be taken not to set the initial cluster membership too close to a strong local minimum of (6).

Selection of the optimal number of clusters is generally not focused on in this study. We assume that K is known in advance or can simply be determined by restarting the algorithm for each plausible K and picking the value that minimizes the total residual. It is worth noting that our method is scalable as it can easily handle data composed of many subgroups, e.g. large unlabeled face image collections.

Regarding step 4 in Algorithm 1, the search is conducted by temporarily moving each point to each cluster other than its current cluster and recording the moves that yield the largest drop in the total residual over the entire dataset. This strategy offers fast convergence. However, to prevent artifacts, such as empty clusters, extra constraints are imposed. First, the minimum number of data points per cluster, s_{min}, is enforced. Secondly, the residual $\|\mathbf{x}_i - \upsilon_j(\mathbf{x}_i)\|^2$ increases with the number of non-zero coefficients in α_j^* (i.e. with cluster size). We compensate for this when assembling basis $\mathbf{\Psi}_{\mathbf{x}_i}$ in (4) by sampling $m = s_{min}$ points several times from each cluster, regardless of its size, and averaging the residual under $\mathbf{\Psi}_{\mathbf{x}_i}$. Therefore, basis for vector $\mathbf{x}_i$ becomes:

$$\mathbf{\Psi}_{\mathbf{x}_i} = [\mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_m^{(1)}, \ldots, \mathbf{x}_1^{(K)}, \ldots, \mathbf{x}_m^{(K)}] \quad , \tag{8}$$

where $\mathbf{x}_j^{(k)}$ denotes j-th vector in the k-th cluster's sample.

It should be noted that compressive projections and reconstructions are more expensive than operations performed in most other clustering approaches, e.g. distance computations in centroid-based methods. However, thanks to introducing constant-size samples for sparsity basis construction, data compression via $\ell 1$-regularized least-squares regression in (2) can be done only once, prior to the main loop. It requires generating a number of random measurement matrices $\mathbf{\Phi}$ in advance and pre-multiplying them by the basis vectors $\mathbf{x}_i$. As point re-assignments and nearest subspace determination involve only relatively cheap residual computations, this dramatically reduces the overall algorithm's execution time at the cost of extra memory requirement.

III. Numerical Results

The proposed compressive clustering algorithm was evaluated through automatic detection of groups of similar objects within four image datasets. Below a brief description of these datasets and the test procedure is given. In Section III-B the obtained clustering results are discussed and compared to those produced using conventional methods.

A. Datasets and Experimental Setup

Four image datasets were used in our experiments [1]:

- ***MPEG-7 shapes*** - Subset of the *MPEG-7* dataset containing various object silhouettes. 10 most consistently looking objects were selected for clustering, each represented by

[1]Datasets can be downloaded from: http://www.dabi.temple.edu/~shape/MPEG7/MPEG7dataset.zip (MPEG-7), http://yann.lecun.com/exdb/mnist/ (MNIST), http://aruta.pl/research/cs/datasets/cars (Cars), and http://vision.ucsd.edu/extyaleb/CroppedYaleBZip/CroppedYale.zip (Ext. Yale Faces).

Dataset	Num. of classes	Images per class	Basis vectors per class	Input dim.	Working dim.
MPEG-7	10	17	8	680	21
MNIST	10	50	20	816	125
Car fronts	12	25	10	2728	115
Car rears	12	25	10	2728	115
Faces	38	15	10	680	153

Table I
EXPERIMENTAL SETUP PARAMETERS WITH RESPECT TO EACH DATASET.

17-20 images scaled to 32×32 px. This easy dataset features limited intra-class and fairly high inter-class variance.

- ***MNIST*** - Large database of handwritten digits from which we took random 10,000 samples, each represented by a 28×28 px gray-scale image. Due to writing style differences, this dataset features high intra-class variance.
- ***Cars*** - Set of 720 well-aligned views of 12 car models, 360 frontal and 360 rear. For a given viewpoint each model is represented by 30 200×100 px images. Small camera's pan and tilt variations were allowed when the pictures were taken. The cars representing the same model differ in terms of body color and general scene illumination.
- ***Extended Yale Faces*** - Database of 168×192 px aligned face images of 38 subjects, split by pose. In our experiments only frontal face images were used, giving 65 images per subject. Different views of the same person exhibit minor facial expression and severe illumination variations.

All images were represented by pyramid Histograms of Oriented Gradients (pHOGs) [17] computed in quad-tree like fashion at first 3-4 levels and flattened into a long 1-D vector.

Each dataset was sampled to keep the input to clusterer of limited size. For each sampling repetition clustering was restarted 10 times with random/*k-means* initialization and the performance over all restarts was averaged. The number of images representing each class in the data was kept constant per dataset. The sample size for sparsity basis construction was computed from (3) assuming $T = s_{min}$. Table I summarizes the experimental setup parameters.

For results validation we compared data partitioning obtained with our algorithm to the gold standard encoded by class labels $\mathbf{y}$ known in advance for each data point, but so far unused. The cluster consistency measure adopted here is based on error matrix – an analogue of confusion matrix, but with decorrelated cluster identifiers along rows and columns. Specifically, for each discovered cluster its intersection with each true class is computed and the maximum intersection is determined. Maxima are then summed and normalized which gives the following cluster quality score:

$$q(c) = \frac{1}{K}\sum_{j=1}^{K}\max_{k=1,\ldots,K}|I_{j,k}| \ , \tag{9}$$

where $I_{j,k} = \{\mathbf{x}_i : c(\mathbf{x}_i) = j \wedge y(\mathbf{x}_i) = k\}$. Note that this approach is in principle different from those that evaluate the obtained clusters based on the data being clustered itself.

Our approach has also been compared to three well-established clustering schemes: *k-means*, the agglomerative algorithm in three best configurations of linkage type and distance function, and Gaussian Mixture. In addition, the first two methods were tested separately in the original feature space and in low-dimensional space obtained via PCA such that the dimensionality matched that used in compressive clustering (see Tab. II in Section III-B). For GMM we fixed the dimensionality to 20 to avoid numerical problems in the EM algorithm.

B. Discussion of the Results

Table II presents intersections of the cluster sets produced by each tested algorithm with ground truth data partitioning. In addition, Figure 1 illustrates the contents of sample clusters found by our algorithm against the contents of the corresponding clusters (with respect to the dominant class) produced by the best alternative method and the ground truth. Class label consistency discrepancies are apparent.

As seen, clusters induced by the proposed algorithm reflect true object categories better than those found using the other methods. However, for the *MPEG-7* dataset it holds only when the compressive clusterer is chained with *k-means* for better initialization. Although the ground truth intersections obtained with our method still seem small for more challenging datasets, it should be noted that the clusters it produces group objects representing on average fewer classes. It is illustrated in Figure 1 and in the sample error matrices from Figure 2 which should be interpreted as: the less dense output (or fewer gray cells in it), the more accurate clustering. This observation suggests that the proposed compressive clustering algorithm is best suited for tasks where external evaluation is the most natural way of assessing how well the internal data structure was captured.

For the last three datasets the clustering schemes based on distance computation expectedly seem to bias towards inter- rather than intra-class commonalities. This is seen for instance in the cluster of "four" digit images in Figure 1 where bias is towards slanting handwriting style.

IV. CONCLUSIONS

In this paper we proposed a novel approach to high-dimensional data clustering, suitable for discovering irregularly distributed object categories. The algorithm is posed as an iterative approximation to the solution of an optimization problem and is based on the compressed sensing paradigm. The goal is to assign each data point to a subset of the remaining points such that the projections of all points onto their resulting linear subspaces jointly minimize the information loss.

Our algorithm was tested against several pre-labeled image collections of varying difficulty with respect to class distributions (from fairly regular to multimodal) and overlap (from clearly separable to highly overlapping). In all cases it showed its ability to capture semantic categories in data more accurately than the conventional methods while using

Dataset	*k-means*		Ward linkage / Euclidean dist.		Weighted linkage / cosine dist.		Average linkage / Spearman dist.		GMM	Compressive clustering	
	full dim.	low dim.	full dim.	low dim.	full dim.	low dim.	full dim.	low dim.	dim = 20	random init.	*k-means* init.
MPEG-7	0.847	0.838	0.859	0.857	0.738	0.918	0.875	0.826	0.741	0.838	**0.962**
MNIST	0.430	0.464	0.460	0.432	0.362	0.424	0.360	0.501	0.486	**0.533**	0.504
Car fronts	0.375	0.380	0.347	0.350	0.327	0.340	0.353	0.640	0.431	**0.773**	0.514
Car rears	0.345	0.368	0.347	0.397	0.253	0.313	0.607	0.673	0.433	**0.765**	0.539
Faces	0.167	0.175	0.179	0.191	0.135	0.218	0.137	0.366	0.191	**0.374**	N/A

Table II
CONSISTENCY OF THE OBTAINED CLUSTERS WITH THE GROUND TRUTH CLASS LABELS FOR DIFFERENT ALGORITHMS TESTED ON OUR DATASETS. BEST QUALITY SCORES FOR EACH DATASET ARE HIGHLIGHTED.

Figure 1. Contents of two sample clusters discovered in the *car fronts* and *MNIST* datasets by the compressive algorithm (left column), the corresponding clusters yielded by the algorithm found best among all tested alternative methods for these datasets (center column), and the ground truth (right column).

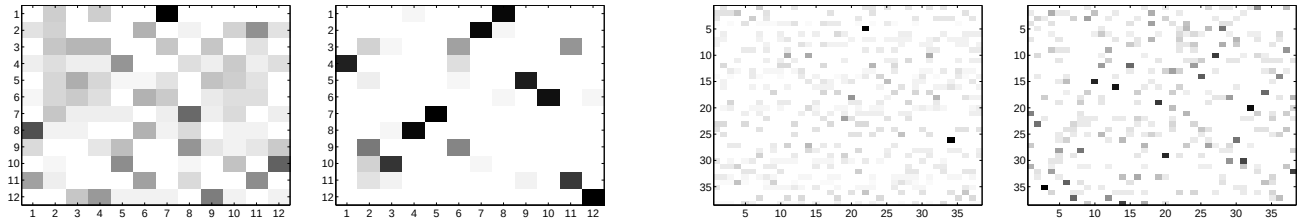

Figure 2. Sample error matrices showing consistency of the clusters discovered in the *car rears* (left) and *Extended Yale Faces* (right) datasets with ground truth class labels. The left matrix in either pair is the output of *k-means* algorithm and the right matrix in either pair was produced by our algorithm.

only a fraction of original information. In the same time it proved to be more invariant to inter-category commonalities.

REFERENCES

[1] R. Sibson, "SLINK: An optimally efficient algorithm for the single-link cluster method," *The Computer Journal*, vol. 16, no. 1, pp. 30–34, 1973.

[2] D. Defays, "An efficient algorithm for a complete link method," *The Computer Journal*, vol. 24, no. 4, pp. 364–366, 1977.

[3] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[4] M. Figueiredo and A. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, 2002.

[5] N. Bouguila and D. Ziou, "High-dimensional unsupervised selection and estimation of a finite generalized Dirichlet mixture model based on minimum message length," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1716–1731, 2007.

[6] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[7] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. of the 2nd Int. Conf. on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[8] D. Comaniciu and P. Meer, "Mean-shift: A robust approach toward feature space analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[9] R. Agraval, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data," *Data Mining and Knowledge Discovery*, vol. 11, no. 1, pp. 5–33, 2005.

[10] K. Kailing, H.-P. Kriegel, and P. Krger, "Density-connected subspace clustering for high-dimensional data," in *Proc. of SIAM Int. Conf. on Data Mining*, 2004, pp. 246–257.

[11] S. Boutemedjet, N. Bouguila, and D. Ziou, "A hybrid feature extraction selection approach for high-dimensional non-Gaussian data clustering," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 8, pp. 1429–1443, 2009.

[12] D. Donoho, "Compressed sensing," *IEEE Trans. on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[13] E. Candés, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

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

[15] Y. Chi and F. Porikli, "Connecting the dots in multi-class classification: From nearest subspace to collaborative representation," in *Proc. of the 25th Int. Conf. on Computer Vision and Pattern Recognition*, 2012, pp. 1–8.

[16] Z. Luo and P. Tseng, "On the convergence of the coordinate descent method for convex differentiable minimization," *Journal of Optimization Theory and Applications*, vol. 72, no. 1, pp. 7–35, 1992.

[17] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of the 18th Int. Conf. on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.