Classification and Clustering via Dictionary Learning with Structured Incoherence and Shared Features

Ignacio Ramirez, Pablo Sprechmann, and Guillermo Sapiro[*]
Electrical and Computer Engineering, University of Minnesota

Abstract

A clustering framework within the sparse modeling and dictionary learning setting is introduced in this work. Instead of searching for the set of centroid that best fit the data, as in k-means type of approaches that model the data as distributions around discrete points, we optimize for a set of dictionaries, one for each cluster, for which the signals are best reconstructed in a sparse coding manner. Thereby, we are modeling the data as a union of learned low dimensional subspaces, and data points associated to subspaces spanned by just a few atoms of the same learned dictionary are clustered together. An incoherence promoting term encourages dictionaries associated to different classes to be as independent as possible, while still allowing for different classes to share features. This term directly acts on the dictionaries, thereby being applicable both in the supervised and unsupervised settings. Using learned dictionaries for classification and clustering makes this method robust and well suited to handle large datasets. The proposed framework uses a novel measurement for the quality of the sparse representation, inspired by the robustness of the ℓ_1 regularization term in sparse coding. In the case of unsupervised classification and/or clustering, a new initialization based on combining sparse coding with spectral clustering is proposed. This initialization clusters the dictionary atoms, and therefore is based on solving a low dimensional eigen-decomposition problem, being applicable to large datasets. We first illustrate the proposed framework with examples on standard image and speech datasets in the supervised classification setting, obtaining results comparable to the state-of-the-art with this simple approach. We then present experiments for fully unsupervised clustering on extended standard datasets and texture images, obtaining excellent performance.

1. Introduction and Basic Formulation

In recent years, sparse representations have received a lot of attention from the signal processing community. This is due in part to the fact that an important variety of signals such as audio and natural images can be well approximated by a linear combination of a few elements (atoms) of some (often) redundant basis, usually called dictionaries [2].

Sparse modeling aims at learning these non parametric dictionaries from the data itself. Several algorithms have been developed for this task, e.g., the K-SVD and the method of optimal directions (MOD). Recent publications in a wide spectrum of signals and applications have shown that this approach can be very successful, leading to state-of-the art results, e.g., in image restoration and denoising, texture synthesis, and texture classification. In the supervised or weakly supervised classification setting, this class of algorithms learn dictionaries from the labeled training dataset and use features of the sparse decomposition of the testing signal for classification (see [14, 17, 22, 34]).

In this paper we propose a framework for clustering datasets that are well represented in the sparse modeling framework with a set of learned dictionaries (see [26] for our earlier work in this direction). Given K clusters, we learn K dictionaries for representing the data, and then associate each signal to the dictionary for which the "best" sparse decomposition is obtained. Note that it is not that each data point belongs to a union of subspaces as for example in [7, 10]. Comparing with block/group sparsity, here a single dictionary (block) is selected per data point, and the point is sparsely represented (subspace) with atoms only from this dictionary.[1] Also in contrast with more classical subspace clustering, data points in the same class can belong to more than one subspace, since each dictionary represents a large number of subspaces (each sparsity pattern defines one subspace). The model is then very rich and nonlinear.

The first building block of the proposed clustering framework is based on considering

$$\min_{\mathbf{D}_i, C_i} \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in C_i} \mathcal{R}(\mathbf{x}_j, \mathbf{D}_i), \tag{1}$$

[*] IR and PS contributed equally.

[1] Sharing atoms between the classes, and therefore having non-empty intersecting subspaces is permitted in our framework as well, see Section 3.1

978-1-4244-6985-7/10/$26.00 ©2010 IEEE

where $\mathbf{D}_i = [\mathbf{d}_1|\mathbf{d}_2 \ldots |\mathbf{d}_{k_i}] \in \mathbb{R}^{n \times k_i}$ is a dictionary of k_i atoms associated with the class C_i, $\mathbf{x}_j \in \mathbb{R}^n$ are the data vectors, and $\mathcal{R}$ is a function that measures how good the sparse decomposition for the signal $\mathbf{x}_j$ under the dictionary $\mathbf{D}_i$ is. In the general case, different dictionaries may have different number of atoms, k_i might be cluster dependent. This problem is closely related with the k-q-flat algorithm that aims at finding the closest k q-dimensional flats to a dataset [30]. However, there are major differences between the two. In particular, the framework here proposed, following the sparse representation approach, considers a large number of flats per class, and does not assume a pre-defined, or even constant across classes, (q) dimension, resulting in a richer space for representing and clustering the signals.

To complete the model, we add a block/dictionary incoherence term, inspired in part by the works on standard sparse coding, e.g., [4, 6, 7, 29], where it was shown that both the speed and accuracy of sparse coding techniques such as soft-thresholding and orthogonal matching pursuit depend on the incoherence between the dictionary atoms. Here we add a term $\mathcal{Q}(\mathbf{D}_i, \mathbf{D}_j)$ that promotes incoherence between the different dictionaries, thereby obtaining a general energy of the form

$$\min_{\mathbf{D}_i, C_i} \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in C_i} \mathcal{R}(\mathbf{x}_j, \mathbf{D}_i) + \eta \sum_{i \neq j} \mathcal{Q}(\mathbf{D}_i, \mathbf{D}_j). \qquad (2)$$

This energy will then lead to the learning of dictionaries optimized to properly represent the corresponding class, due to $\mathcal{R}$, while at the same time being weak for the other classes, due to the term $\mathcal{Q}$. We will later show how classes can still share atoms, an important property for classification algorithms [28], and a unique characteristic of our proposed model. Note that in contrast with prior work on dictionary learning for classification, this novel cross dictionary learning term $\mathcal{Q}$ is independent of the data, is intrinsic to the dictionaries being learned, thereby rendering itself also to the case of unsupervised or semi-supervised classification and clustering. For the experiments in this paper we use the terms $\mathcal{Q}(\mathbf{D}_i, \mathbf{D}_j) = \|\mathbf{D}_i^T \mathbf{D}_j\|_F^2$, where the subscript F denotes Frobenius norm.[2]

We propose a measurement $\mathcal{R}$ for the quality of the sparse representation that naturally takes into account both the reconstruction error and the sparseness (complexity) of the representation on the corresponding learned dictionary. Such measurement can be applied to image patches directly or to image features, e.g., SIFT as in [34]. In practice this measurement has shown enormous discrimination power. To further show this, we performed experiments in the supervised classification setting using labeled data; we first learned a dictionary for each class (with the incoherence promoting term $\mathcal{Q}$), and then classified each testing signal according to this measure. This very simple approach gives results comparable with the state-of-the-art for several benchmark datasets. Thereby, as a by-product of our proposed clustering framework, we obtain a very simple and efficient supervised classification technique as well.

In the unsupervised clustering case, the initialization is very important for the success of the algorithm. Due to the cost associated with the procedure, repeating random initializations is practically impossible. Thus a "smart" initialization is needed. We propose an approach that combines sparse coding with spectral clustering [19], and is applicable to large datasets.

Ideas related to the ones here proposed were previously employed for subspace clustering [8, 11, 24], clustering using the so-called *ℓ_1-graph* by Huang and Yan (see description in [33]), and label propagation [3]. In contrast with our proposed dictionary learning framework, these works model all the data points in a given class as belonging to the same unique subspace, while we model them as "belonging" to the same dictionary, a richer non-linear model since each subset of atoms from the dictionary represents a different subspace. Moreover, these very inspiring approaches all use the data itself as dictionary, sparsely representing every data point as a linear combination of the rest of the data. Such representation is computationally expensive (virtually unusable for datasets of thousands of points). In addition, the large redundancy and coherence expected from using the data itself as dictionary is prompt to make the sparse coding very unstable: as mentioned above, it is well known that such coding techniques strongly depend on the internal coherence of the dictionary. Furthermore, the performance of these methods decreases when the number of clusters grows. We propose as part of our framework a method to bypass this problem that divides the clustering problem into several binary ones. In a natural way, we use the proposed energy function to decide which partition to choose. Such binary division framework is not so natural for these other related clustering methods.

In Section 2 we summarize the main ideas of sparse coding and dictionary learning. In Section 3 we define the measure $\mathcal{R}$ and analyze its discriminative power providing examples of supervised classification. In Section 4 we present the proposed clustering algorithm, together with theoretical guarantees and experimental results. Finally, we conclude the paper in Section 5.

2. Sparse Coding and Dictionary Learning

Sparse coding means to represent a signal as a linear combination of a few atoms of a given dictionary. Mathematically, given a signal $\mathbf{x} \in \mathbb{R}^n$ and a dictionary $\mathbf{D} \in \mathbb{R}^{n \times k}$, the sparse representation problem can be stated as

[2] We can also easily add internal incoherence between the atoms of each dictionary [23], in order to further stabilize not only the dictionary selection but the particular atoms in the corresponding dictionary. This is done here for the initialization step.

$\min_{\mathbf{a}} \|\mathbf{a}\|_0$, s.t. $\mathbf{x} = \mathbf{D}\mathbf{a}$, where $\|\mathbf{a}\|_0$ is the ℓ_0 pseudo-norm of the coefficient vector $\mathbf{a} \in \mathbb{R}^k$, the number of non-zero elements. As minimizing ℓ_0 is NP-hard, a common approximation is to replace it with the ℓ_1-norm. In the noisy case the equality constraint must be relaxed as well. An alternative then is to solve the unconstrained problem,

$$\min_{\mathbf{a}} \|\mathbf{x} - \mathbf{D}\mathbf{a}\|_2^2 + \lambda \|\mathbf{a}\|_1, \tag{3}$$

where λ is a parameter that balances the tradeoff between reconstruction error and sparsity. It is a well known fact that the ℓ_1 constraint induces sparse solutions for the coefficient vectors $\mathbf{a}$. Furthermore, this is a convex problem that can be solved very efficiently using for example the LARS-Lasso algorithm [5]. This alternative has also been shown to be more stable than the ℓ_0 approach in the sense that in the latter, small variations in the input signal can produce very different active sets (the set of non-zero coefficients in $\mathbf{a}$, or selected atoms from $\mathbf{D}$).

Now, what about the actual dictionary $\mathbf{D}$? State-of-the-art results have shown that it should in general be learned from data. Given a set of signals $\{\mathbf{x}_i\}_{i=1\ldots m}$ in $\mathbb{R}^n$, the goal is to find a dictionary $\mathbf{D} \in \mathbb{R}^{n \times k}$ such that each signal in the set can be represented as a sparse linear combination of its atoms. In this work we use a variation of [16], where learning the dictionary is done by seeking a (local) solution to the following optimization problem,

$$\min_{\mathbf{D}, \{\mathbf{a}_i\}_{i=1,\ldots,m}} \sum_{i=1}^{m} \|\mathbf{x}_i - \mathbf{D}\mathbf{a}_i\|_2^2 + \lambda \|\mathbf{a}_i\|_1, \tag{4}$$

while restricting the atoms to have norm less than one. The optimization is carried out using an iterative approach that is composed of two (convex) steps: the sparse coding step on a fixed $\mathbf{D}$ and the dictionary update step on fixed $\mathbf{a}$.

3. The Sparse Representation Quality $\hat{\mathcal{R}}$ and Supervised Classification

A common approach when using dictionaries for classification is to train class specific dictionaries using labeled data and then assign each testing signal to the class for which the best reconstruction is obtained [17, 22]. The measure employed for this task is often the reconstruction error, $\mathcal{R}(\mathbf{x}, \mathbf{D}) = \|\mathbf{x} - \mathbf{D}\mathbf{a}\|_2^2$, where $\mathbf{a}$ is the optimal coefficient vector in the sparse coding. While this strategy leads to very good results, it does not take into account the actual sparsity of the reconstruction. Suppose that we have two dictionaries for which almost the same reconstruction error is obtained, but one of them requires double the atoms than the other. In such a situation one would rather select the dictionary that gives the sparsest solution (simplest, following Akaike's Information Principle [1]), even if the reconstruction error is slightly larger.

dataset	proposed	data	A	B	C	SVM	k-NN
MNIST	1.26	1.35	3.41	**1.05**	-	1.4	5.0
USPS	3.98	4.14	**3.56**	4.38	6.05	4.2	5.2
ISOLET	**3.01**	3.34	4.3	3.4	-	3.3	8.7

Table 1. *Error rate (in percentage) for the algorithm discussed in Section 3. We present comparisons with recently published approaches (results taken from the corresponding papers). The "data" column corresponds to using our discrimination function with dictionaries formed with the whole training dataset.* MNIST*: (A) is the best reconstructive method presented in [18], while (B) is the best discriminative one.* USPS*: (A) is the best reconstructive and (B) is the best discriminative method, both reported in [18]. (C) is the best result obtained in [12] (only* USPS *available).* ISOLET*: (A) is the supervised k-q-flats and (B) is the k-metrics in [27]. We also compare with an* SVM *with Gaussian kernel and the Euclidean k-NN.*

In practice, this problem can be addressed using a small pre-defined sparsity level L in an ℓ_0 approach. This strategy is not longer valid when the convex relaxation (3) is employed (such relaxation is critical for classification tasks requiring robustness and stability). In this situation, comparing the reconstruction errors alone has little meaning. We propose then to use the actual cost function in (3) as a measure of performance, as in the dictionary learning (4), $\hat{\mathcal{R}}(\mathbf{x}, \mathbf{D}) = \min_{\mathbf{a}} \|\mathbf{x} - \mathbf{D}\mathbf{a}\|_2^2 + \lambda \|\mathbf{a}\|_1$. This alternative takes into account both the reconstruction error and the complexity of the sparse decomposition. The reconstruction error measures the quality of the approximation while the complexity is measured by the ℓ_1 norm of the optimal $\mathbf{a}$.

Let $\mathbf{X}_i$, $i = 1, \ldots, K$, be a collection of K (labeled) classes of signals and $\mathbf{D}_i$ the corresponding dictionaries trained for each of them independently following for example (4). This gives, for each class, a (reconstructive) dictionary unaware of the task (classification/clustering) and of the data in the other classes. Thereby, as detailed in the introduction, it is more appropriate to add the dictionary incoherence $\mathcal{Q}(\mathbf{D}_i, \mathbf{D}_j)$, and the proposed optimization is

$$\min_{\{\mathbf{D}_i, A_i\}_{i=1\ldots K}} \sum_{i=1}^{K} \left\{ \|\mathbf{X}_i - \mathbf{D}_i \mathbf{A}_i\|_2^2 + \lambda \sum_{j=1}^{m_i} \|\mathbf{a}_i^j\|_1 \right\} + \eta \sum_{i \neq j} \|\mathbf{D}_i^T \mathbf{D}_j\|_F^2. \tag{5}$$

Here we used the standard notation $\mathbf{A}_i = [\mathbf{a}_i^1 \ldots \mathbf{a}_i^{m_i}] \in \mathbb{R}^{k_i \times m_i}$, each column $\mathbf{a}_i^j$ is the sparse code corresponding to the signal $j \in [1..m_i]$ in class i. Note that the first term in the optimization is as in (4), where each dictionary is optimized for the data from its own class. The second term provides the coupling. In contrast with works such as [17], the coupling is between the dictionaries, the labeled data points do not form part of this term, thereby this can be used also in the non-supervised learning process, see next section.

Once the dictionaries have been learned, the class $\hat{j}_0$ for a given new signal $\mathbf{x}$ is found by solving $\hat{j}_0 =$

$\arg\min_{j=1,\ldots,K} \hat{\mathcal{R}}(\mathbf{x}, \mathbf{D}_j)$.[3] This procedure is very simple and its few parameters can be found via cross-validation.

3.1. Sharing Atoms

In practice, it turns out that even though we impose incoherence in the dictionaries, atoms representing common features in all classes tend to appear repeated almost exactly in dictionaries corresponding to different classes. Being so common, these atoms are used often and their associated reconstruction coefficients have a high absolute value $|\mathbf{a}_r|, r \in \{1, \ldots, k_i\}$, thus making the reconstruction costs $\hat{\mathcal{R}}(\mathbf{x}, \mathbf{D}_i)$ similar. By ignoring the coefficients associated to these common atoms when computing $\hat{\mathcal{R}}$, we can improve the discriminatory power of the system. The natural way to detect such atoms is to inspect the already available $\mathbf{D}_i^T \mathbf{D}_j$ matrices, whose absolute values represent the inner products between atoms. In the following experiments, a threshold of 0.95 consistently improves the results, sometimes significantly. Note that this procedure accounts for allowing classes to share features [28], and the corresponding subspaces to have intersections, in contrast for example with [8]. Figure 1 illustrates examples of automatically learned shared atoms in the task of learning to classify digits from the MNIST dataset. See [9] for the selection of features (atoms) for parametric dictionaries.

3.2. Experimental Results

We first test this simple classification method with standard datasets, the MNIST and USPS digit datasets and the ISOLET data that consists of 617 audio features extracted from 200 speakers saying each letter of the alphabet twice. We used in every case the usual training/testing split. In Table 1 we present the obtained results. We compare our results with several much more sophisticated classification algorithms. The results obtained are comparable and sometimes even better. We also compare with the standard Euclidean k-NN and with SVM with a Gaussian kernel. In all our experiments we used a penalty parameter $\lambda = 0.1$. The size of the dictionary depends on the number of training samples as well as the intrinsic complexity of the data. For MNIST, which has many samples, our best results were obtained with $k = 800$. In contrast, USPS and ISOLET have much less samples and more variability, leading to a much smaller dictionaries of size $k = 80$ and $k = 60$ respectively. These already state-of-the-art results can be further improved for example using the $\hat{\mathcal{R}}_i$ in an SVM.

One could think of using the whole training datasets as dictionaries for each class as with the approaches mentioned in the introduction [8, 24, 33]. In that case, in all our experiments the error rates obtained are not better than the ones reported in Table 1. Using the data as dictionaries has the additional disadvantage that the computational cost of the classification becomes prohibitive,[4] and the method is highly susceptible to label errors due to the high coherence of the "dictionary."

Finally, we illustrate the discrimination power of the measure $\hat{\mathcal{R}}$ in a more challenging scenario using images from the Grasz02 dataset [20]. We address the object detection task by learning dictionaries for the local SIFT descriptors of an object class.

We chose the "bike" class from the Graz02 as an example, and test our proposed framework in two different weakly supervised settings. In the first setting, along with the training images, we provide the algorithm with a bounding box enclosing the bikes present in each of these images. In the second, the only supplied information is whether a bike is present or not in each of the training images. Clearly, the second case is more challenging. On each image we extract 128 dimensional SIFT descriptors from patches of 32×32 pixels computed over a grid with spacing of 4 pixels. For the first setting, we randomly pick $300,000$ SIFT descriptors from inside and outside of the bounding box respectively, and learn corresponding dictionaries with 500 atoms each. In the second setting, the bike and background dictionaries were learned from all the patches extracted from images marked as either containing a bike or purely background respectively.

In both cases, the dictionary for the class "bike" was learned iteratively, keeping the 90% of the descriptors that were more clearly assigned to the class "bike" at each iteration. This allows us to gradually discard background descriptors labeled as "bike." The choice of training and testing images was performed as it is usual for this dataset, where the first 300 images are split in two, the odd images for training, and the even images for testing.

Since classified patches overlap, each pixel in the image has several possible energy values $\hat{\mathcal{R}}$ for each of the two dictionaries, one per patch covering it. This spatial redundancy helps the algorithm to determine a more accurate energy value at the pixel level by means of a simple spatial average, with a Gaussian kernel, giving more weight to patches in which the pixel is closer to the center. Using a Gaussian regularization on the energy images has proven to improve the results. This is in part due to the way the ground truth masks are defined, Figure 2. The wheels are labeled as belonging to the class "bike" while most of the time one can see the background behind them. A strategy that considers the features globally or at several scales would help [13, 34], but this is beyond the scope of this example.

In Figure 2 we show the detection results obtained with this framework. We also show the corresponding precision vs. recall curve for the whole testing set. The results are

[3] We actually obtain more than this information, since for each $\mathbf{x}$ we compute all the $\hat{\mathcal{R}}$s for all the K classes, and thereby can provide a soft classification with probabilities, or a feature vector for an SVM.

[4] The cost of learning the dictionaries in our approach is off-line.

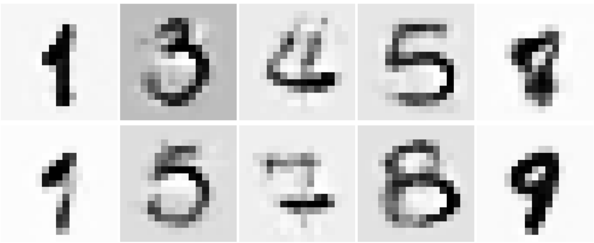

Figure 1. *Atoms discarded due to excessive coherence. From left to right: 1 vs. 9, 3 vs. 5, 4 vs. 7, 5 vs. 8, 8 vs. 9. Notice how these atoms have learned features shared between different classes.*

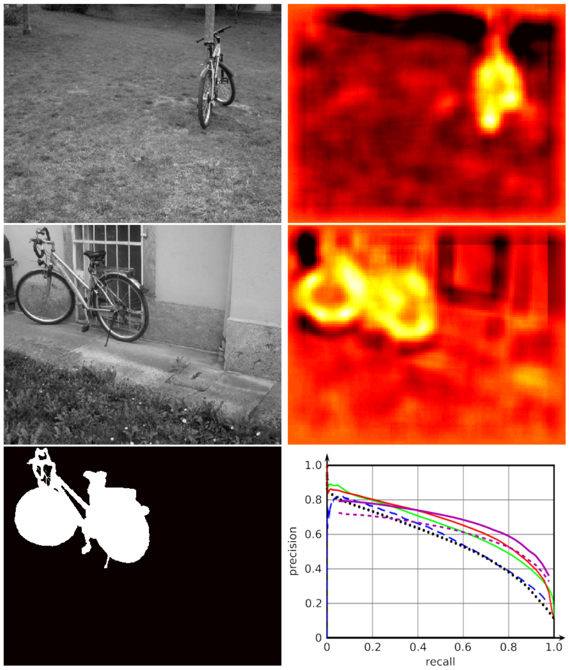

Figure 2. *Bike detection on the Graz dataset using the measure $\hat{\mathcal{R}}$. The two topmost rows show the obtained detection for two sample images. The colored area corresponds to the regions for which the representation energy using the bike dictionary is smaller than the background one. The lighter the color, the more "bike-like" is the pixel. Bottom left: shows the ground truth for the middle row. Bottom right: precision vs. recall curve for several algorithms [21] (blue,dashed), [31] (black,dotted), [17] (red and green), and the proposed algorithm, using a bounding box (magenta,solid), and weakly supervised (magenta,dashed).*

very good, comparable to state-of-the-art, considering that we are using one single dictionary to categorize each of these highly complex categories. In this object localization application, the cancelation of atoms of high coherence has a crucial role because of the similarities that both classes have at the local level. If one uses directly dictionaries trained independently for each class, then most of the diagonal vertexes on the image tend to be classified as "bikes" and the opposite happens with horizontal and vertical edges, which are very frequent in urban environments.

4. Dictionary Learning for Clustering

We now proceed to extend the above dictionary learning and sparse coding frameworks to unsupervised clustering. Given a set of signals, $\{\mathbf{x}_j\}_{j=1\dots m}$ in $\mathbb{R}^n$, and the number of clusters/classes, K,[5] we want to find the set of K dictionaries $\mathbf{D}_i \in \mathbb{R}^{n\times k_i}$, $i = 1,\dots,K$, that best represents the data. We formulate this as an energy minimization problem of the form of Equation (2), and use the measure proposed in Section 3,

$$\min_{\mathbf{D}_i, C_i} \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in C_i} \min_{\mathbf{a}_i^j} \|\mathbf{x}_j - \mathbf{D}_i \mathbf{a}_i^j\|_2^2 + \lambda \|\mathbf{a}_i^j\|_1 + \eta \sum_{i\neq j} \|\mathbf{D}_i^T \mathbf{D}_j\|_F^2, \quad (6)$$

where as before, the atoms of all the dictionaries are restricted to have unit norm. In contrast with (5), class assignments are unknown, and the optimization is carried out iteratively using a Lloyd's-type algorithm: *Assignment step:* The dictionaries are fixed and each signal is assigned to the cluster for which the best representation is obtained: $C_{j_0} := \left\{\mathbf{x} : \hat{\mathcal{R}}(\mathbf{x}, \mathbf{D}_{j_0}) \le \hat{\mathcal{R}}(\mathbf{x}, \mathbf{D}_i)\ \forall i = 1,\dots,K\right\}$ (omitting the contribution of shared atoms). *Update step:* The new dictionaries are computed fixing the assignments found in the previous step. This is the dictionary learning problem (4), with the addition of the incoherence term.

The algorithm stops when the relative change in the energy is less than a given constant. In practice few iterations are needed to reach good results. While the energy is being reduced at every step, there is no guarantee of arriving to a global minimum. In this setting, repeated initializations are computationally very expensive, thus a good initialization is required. This is explained next.

4.1. Initialization: Spectral Clustering Meets Dictionary Learning

The initialization for the algorithm presented in the previous section can be given as a set of K dictionaries or as an initial partition of the data, this is the C_i sets. We propose two closely related algorithms one corresponding to each of these two alternatives. In both cases the main idea is to construct a similarity matrix and use it as the input for a spectral clustering algorithm [32].

Let $\mathbf{D}_0 \in \mathbb{R}^{n\times k_0}$ be an initial global dictionary trained (with internal incoherence) to reconstruct the data for the whole (unlabeled) set $X := [\mathbf{x}_1,\dots,\mathbf{x}_m]$. For each signal $\mathbf{x}_j$ we have the corresponding sparse representation $\mathbf{a}_j$. Let us define $\mathbf{A} = [\mathbf{a}_1,\dots,\mathbf{a}_m] \in \mathbb{R}^{k_0\times m}$. Two signals belonging to the same cluster are expected to have decompositions that use similar atoms. Thus one can measure the similarity of two signals by comparing the corresponding sparse

[5] When K is over-estimated, a micro-detailed partition is observed.

representations. Inversely, the similarity of two atoms can be determined by comparing how many signals use them simultaneously, and how they contribute, in their sparse decomposition. We compute two matrices representing each one of these cases respectively:

Clustering the signals: Construct a similarity matrix $\mathbf{S}_1 \in \mathbb{R}^{m\times m}$, $\mathbf{S_1} := |\mathbf{A}^T\mathbf{A}|$.

Clustering the atoms: Construct a similarity matrix $\mathbf{S}_2 \in \mathbb{R}^{k_0\times k_0}$, $\mathbf{S_2} := |\mathbf{A}\,\mathbf{A}^T|$.

In both cases the similarity matrix obtained is positive semidefinite and can be associated with a graph, $G_1 := \{\mathbf{X}, \mathbf{S}_1\}$ and $G_2 := \{\mathbf{D}, \mathbf{S}_2\}$, where the data or the atoms are the sets of vertexes with the corresponding $\mathbf{S}_i$ as edge weights matrixes. This graph is partitioned using standard spectral clustering algorithms to obtain the initialization for the algorithm described in the previous section.

As we mentioned before, G_1 is closely related with the ℓ_1-graph. In that case, the weights of the graph are determined using the sparse decomposition of the signals with the data itself as a dictionary. When the number of signals m is large, the computational cost of constructing the similarity matrix is too expensive. Also the spectral clustering algorithm requires the computation of the largest singular values (and corresponding singular vectors), which is also computationally demanding when m is large (although not so demanding if only a few eigenvectors are needed). In the case of G_2, clustering the atoms bypasses these difficulties, the size of $\mathbf{S}_2$ depends on the significantly smaller size of the initial dictionary k_0. This parameter does not depend on the amount of data, it just needs to be large enough to model it properly, and is often just in the hundreds. Note that the obtained sub-dictionaries may have different cardinalities (different k_i), reflecting different complexities of the associated clusters.

When the number of clusters, K, is large, the performance of the initial clusterization decreases. We propose a more robust initialization. Starting with the whole set as the only partition, at each iteration we subdivide in two sets each of the current partitions, keeping the division that produces the biggest decrease in the cost energy defined in Equation (6). The procedure stops when the desired number of clusters is reached. This can be applied for any of the two graphs presented in this section, and such partition is consistent with the energy driving the clustering.

4.2. Theoretical Guarantees

In this Section we show that, under certain ideal conditions, one can prove that the initialization step presented in the previous section produces a perfect clustering of the data. Because this assumptions do not hold in general with real data, the result of the initial step does not always give a correct clustering, but it gives a very good first approximation that is be later refined by the iterative step.

Dataset	k-means	$\eta = 0$	$\eta \neq 0$
MNIST	21.2	6.9	**3.0**
USPS	22.3	2.9	**2.0**
ISOLET	20.0	6.0	**1.5**
Brodatz(x2)	-	2.5	**0.4**

Table 2. *Error rate (in percentage) for the clustering algorithm discussed in Section 4. In* MNIST *and* USPS *we used digits form 0 to 5 and for* ISOLET *we used the last 6 letters. We also tested clustering combinations of 2 randomly chosen Brodatz textures. In this case, the result is the average performance over 10 random realizations. In all cases, the results are shown with ($\eta \neq 0$), and without ($\eta = 0$) added incoherence.*

Following the ideas presented in [8], let us consider the ideal situation in which every signal in the K clusters can be exactly reconstructed as a sparse linear combination of the atoms of a dictionary and that the subspace that they span (using all the atoms) are independent, this is, that their sum is direct. Let us call those subspaces S_i, $i = 1, \ldots, K$. Now, assume that the initial dictionary is composed of K (redundant) sub-dictionaries, $\mathbf{D}_0 = [\mathbf{D}_1, \ldots, \mathbf{D}_K]$, one corresponding to each cluster in the dataset. For simplicity we assume that the atoms of the dictionary $\mathbf{D}_0$ are ordered but this is not required for proving any of the results discussed here.

Then, given a vector $\mathbf{x}$ belonging to one of the subspaces, it is easy to show that the optimal $\mathbf{a}$ in the ℓ_1-relaxation of ℓ_0 with this $\mathbf{D}_0$, will use only atoms from the correct block of the initial dictionary, producing K connected components in both graphs $\mathbf{G}_1$ and $\mathbf{G}_2$. In this situation a spectral clustering technique will successfully separate the clusters [32].

The hypothesis from [8] that the subspaces span independent subspaces is very strong. In practice, different clusters very often have non trivial intersections. This is exactly what is tackled by not considering highly coherent atoms in the comparison of the quality of the sparse representations presented in Section 3.1. One can still prove that the solution to the ℓ_0 problem will pick atoms from the correct block for a given $\mathbf{x} \in S_i$ if the atoms of the dictionaries that compose $\mathbf{D}_0$ satisfy $\max_d ||\mathbf{D}_i^\dagger \mathbf{d}||_1 < 1$, where $\mathbf{D}_i^\dagger$ is the Moore-Penrose pseudoinverse of $\mathbf{D}_i$ and $\mathbf{d}$ is any atom of the dictionaries $\mathbf{D}_j$ with $j \neq i$. This condition is similar to the one required for the exact recovery of the orthogonal matching pursuit algorithm [29]. It is related to the incoherence between atoms belonging to different dictionaries. The proof can be made following similar ideas to the ones used in that case, and is here omitted due to space limitations.

4.3. Clustering Results

We now apply the proposed algorithm to several clustering problems and texture segmentation. We first clustered the digits form 0 to 5 ($K = 6$) from the testing set of MNIST and the training set of USPS (ignoring the labels, of course). We also clustered the last six letters of ISOLET ($K = 6$), combining the standard training and testing sets. We fur-

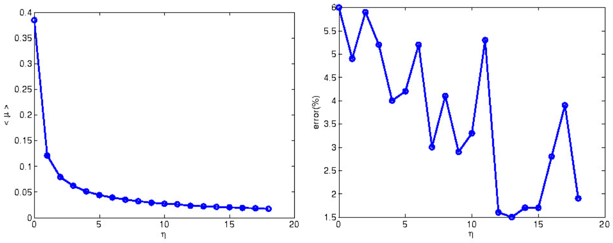

Figure 3. *Effect of incoherence in clustering performance of the* ISOLET *database. Left: average incoherence between all the dictionaries vs. η. Right: classification error vs. η. Note that, due to the small size of this dataset, fluctuations of $\pm 1\%$ such as the ones here observed are common.*

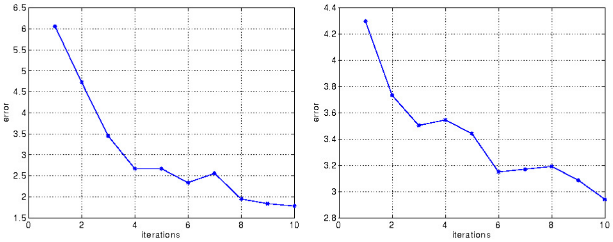

Figure 4. *Classification error vs. number of iterations for the clustering algorithm when applied to the* ISOLET *(left) and* USPS *(right) datasets.*

ther applied the clustering scheme to the combined patches of randomly chosen samples of 2 textures from the Brodatz database. The results are reported in Table 2 for different values of the incoherence penalty term η. The size of the initial dictionaries are $k = 120$ for USPS , $k = 300$ for MNIST and $k = 90$ for ISOLET. The dictionaries representing each cluster are of size 60, 25 and 15 respectively. The initial clustering of the data was done using spectral clustering on the graph G_1. In all the cases involving images, the atoms learned for each cluster were visually identifiable with the classes they represented.

Effect of imposing incoherence: As can be see in Table 2, encouraging incoherence in the dictionaries is of paramount importance, first to the initial dictionaries as internal incoherence, and then between dictionaries during the iterative clustering. To further understand this effect, we show in Figure 3 how the strength of the incoherence term, controlled by the parameter η, affects the reconstruction error. Also shown is the actual average incoherence obtained between the cluster dictionaries, that is the average value of $|\mathbf{d}_i^T \mathbf{d}_j|$ between all possible pairs of atoms $\mathbf{d}_i$ and $\mathbf{d}_j$ from all the dictionaries involved.

Effect of the iterative clustering: In Figure 4 we show an example of how the classificaion error is monotonically reduced for succesive iterations of the proposed algorithm, thus empirically assessing its stability.

Texture segmentation: We also apply our clustering algorithm for the texture segmentation problem. The goal is to assign each pixel on an objective image to one of K possible textures. The approach is related to the one used in [22] for the supervised case. Overlapping 16×16 patches from the original images and used as input signals.

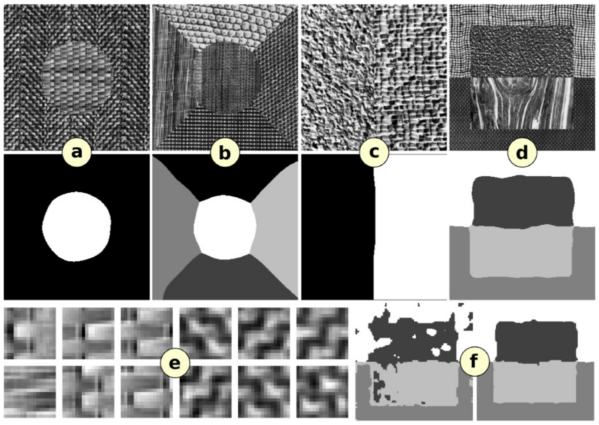

Figure 5. *Texture segmentation results on mosaics from the Brodatz database. (a)-(d) above are the mosaics and below the respective segmentation. The missclassification rates are 1.75%, 4.25%, 0.25% and 3.4% respectively. In (e) we show sample atoms from the final cluster dictionaries for the textures in (a). The texture in the circle required $k_1 = 82$ atoms, while the other one received $k_2 = 118$, which goes along with the intuition of larger complexity for this texture. (f) shows the first and third iteration of the iterative clustering algorithm.*

After each iteration (that is, before recomputing the dictionaries), we obtain K different energy values for each patch, each corresponding to a candidate texture class. Following the same strategy than in the object detection example of Section 3, we use the energy value obtained for each patch to construct K different "energy images," and combine such energies to obtain the pixel-wise classification.

In Figure 5 we show some of the results. The number of patches extracted was on the order of several thousands, so the initialization with G_2 was applied. The algorithm gave sub-dictionaires that have a cardinality that intuitively reflects the complexity of the corresponding texture (in other words, k_i was not constant). We got very low rates of missclassified pixels, for example in Figure 5(c), where we obtained 0.25%, which is better than the 0.37% obtained in [17] for the supervised case (which was, as far as we know, the best reported result in the literature for that image).

5. Concluding Remarks

A framework for classification and clustering based on dictionary learning and sparse representations was introduced in this paper. The basic idea is to simultaneously learn a set of dictionaries that optimally represent each one of the classes. Toward this goal, we introduced a new measurement of representation quality, a new term that promotes incoherence between the dictionaries, and an initialization procedure that combines sparse coding, dictionary learning, and spectral clustering. The clusters are allowed to share atoms. The obtained model is much richer than standard subspace clustering algorithms, since multiple subspaces represent a given class, and classes can have intersecting subspaces. Soft-clustering can be obtained as

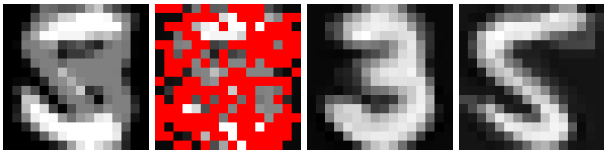

Figure 6. *Example of the recovered digits from a mixture (left) with 60% of missing components (red pixels in second figure), each digit is sparsely represented in its own learned dictionary, and the signal is the result of the sum of two digits (mixture of two dictionaries). As with the case in this paper with signals composed form a single dictionary, the technique described in [25] finds from the corrupted mixture, the correct classification classes (dictionaries) and the reconstruction (last two images).*

well in this framework, and the experimental results can be further improved using these soft measures in an SVM.

While we considered signals sparsely represented each one by a single dictionary, this can be extended to signals resulting from mixtures of dictionaries [25], Figure 6.

Acknowledgments: Work supported by ONR, NGA, NSF, and ARO. Code follows [15].

References

[1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19, 1974.

[2] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Rev.*, 51(1):34–81, 2009.

[3] H. Cheng, Z. Liu, and J. Yang. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In *ICCV*, 2009.

[4] I. Daubechies, M. Defrise, and C. D. Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. on Pure and Applied Mathematics*, 57:1413–1457, 2004.

[5] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[6] M. Elad. Optimized projections for compressed-sensing. *IEEE Trans. SP*, 55(12):5695–5702, Dec. 2007.

[7] Y. C. Eldar and M. Mishali. Robust recovery of signals from a structured union of subspaces. *IEEE Trans. IT.*, 2009.

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

[9] K. Etemad and R. Chellappa. Separability-based multiscale basis selection and feature extraction for signal and image classification. *IEEE Transactions on Image Processing*, 7:1453–1465, 1998.

[10] A. Ganesh, Z. Zhou, and Y. Ma. Separation of a subspace-sparse signal: Algorithms and conditions. In *ICASSP*, volume 14, april 2009.

[11] B. Gowreesunker and A. H. Tewfik. A novel subspace clustering method for dictionary design. In *ICA, Lecture Notes in Computer Science*, pages 34–41. Springer, 2009.

[12] K. Huang and S. Aviyente. Sparse representation for signal classification. In *NIPS*, 2006.

[13] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178, Washington, DC, USA, 2006. IEEE Computer Society.

[14] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *NIPS*, volume 19. 2007.

[15] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:16–60.

[16] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *ICML*, 2009.

[17] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *CVPR*, 2008.

[18] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *NIPS*, volume 21, pages 1033–1040. 2009.

[19] A. Y. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, volume 14. 2002.

[20] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer. Generic object recognition with boosting. *IEEE Trans. on PAMI*, 28(3), March 2006.

[21] C. Pantofaru, G. Dorko, C. Schmid, and M. Hebert. Combining regions and patches for object class localization. In *The Beyond Patches Workshop. CVPR*, 2006.

[22] G. Peyré. Sparse modeling of textures. *Journal of Mathematical Imaging and Vision*, 34:17–31, May 2009.

[23] I. Ramirez, F. Lecumberry, and G. Sapiro. Universal priors for sparse modeling. In *Int. Workshop on Computational Advances in Multi-Sensor Adaptive Proc.*, December 2009.

[24] S. R. Rao, R.Tron, R. Vidal, and Y. Ma. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In *CVPR*, 2008.

[25] P. Sprechmann, I. Ramirez, G. Sapiro, and Y. Eldar. Collaborative hierarchical sparse modeling. In *Annual Conference on Information Sciences and Systems*, March 2010.

[26] P. Sprechmann and G. Sapiro. Dictionary learning and sparse coding for unsupervised clustering. In *Proc. ICASSP*, Mach 2010.

[27] A. Szlam and G. Sapiro. Discriminative k-metrics. In *ICML*, 2009.

[28] A. Torralba, K. Murphy, and W. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Trans. PAMI*, 29(5):854–869, 2007.

[29] J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. IP*, 50:2231–2242, 2004.

[30] P. Tseng. Nearest q-flat to m points. *J. Optim. Theory Appl.*, 105(1):249–252, 2000.

[31] T. Tuytelaars and C. Schmid. Vector quantizing feature space with a regular lattice. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, 2007.

[32] U. von Luxburg. A tutorial on spectral clustering. *CoRR*, abs/0711.0189, 2007.

[33] J. Wright, Y. Ma, J. Mairal, G. Spairo, T. Huang, and S. Yan. Sparse representations for computer vision and pattern recognition. In *Proc. IEEE*, 2010, to appear.

[34] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.