A Review of Sparsity-based Clustering Methods

Article *in* Signal Processing · February 2018

DOI: 10.1016/j.sigpro.2018.02.010

2 authors, including:

Yigit Oktar
Izmir University of Economics
19 PUBLICATIONS **39** CITATIONS

A review of sparsity-based clustering methods

Yigit Oktar[1], Mehmet Turkan[2]

Department of Computer Engineering
Department of Electrical and Electronics Engineering
Izmir University of Economics
Izmir, Turkey

Abstract

In case of high dimensionality, a class of data clustering methods has been proposed as a solution that includes suitable subspace search to find inherent clusters. Sparsity-based clustering approaches include a twist in subspace approach as they incorporate a dimensionality expansion through the usage of an overcomplete dictionary representation. Thus, these approaches provide a broader search space to utilize subspace clustering at large. However, sparsity constraint alone does not enforce structured clusters. Through certain stricter constraints, data grouping is possible, which translates to a type of clustering depending on the types of constraints. The dual of the sparsity constraint, namely the dictionary, is another aspect of the whole sparsity-based clustering methods. Unlike off-the-shelf or fixed-waveform dictionaries, adaptive dictionaries can additionally be utilized to shape the state-model entity into a more adaptive form. Chained with structured sparsity, adaptive dictionaries force the state-model into well-formed clusters. Subspaces designated with structured sparsity can then be dissolved through recursion to acquire deep sparse structures that correspond to a taxonomy. As a final note, such procedure can further be extended to include various other machine learning perspectives.

Keywords: Clustering, Sparse Representations, Structured Sparsity, Deep Sparse Structures

1.Introduction

Clustering is an approach to unsupervised learning. There is no labeling required, unlike classification tasks. In broad terms, clustering can be expressed as *exploring the unknown*. The wide range of clustering applications includes search engines, social networks, visual tasks such as image segmentation, and DNA analysis. Search engines need to cluster information in order to be able to retrieve relevant data in the times of querying. Social networks inherently appear in a clustered nature. Image segmentation is a visual application of clustering. Not surprisingly, molecular biology is a promising domain for clustering applications, due to its aim of discovering the unknown world.

Clustering can simply be defined as the task of grouping entities in terms of a similarity measure. Here, the critical issue is to understand what is meant by "similar". Similarity is in a sense the inverse of a distance metric between two entities. The shorter the distance, the more similar the entities, and vice versa. It is important hence to note that, clustering results will be crucially dependent on the similarity notion chosen. A conventional distance metric is the squared Euclidean distance between two data points $\mathbf{x}$ and $\mathbf{y}$ which is defined as $dist(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$. Many other similarity measures, e.g., [1, 2], could be utilized to tackle the broad range of domain specific clustering problems.

Clustering methods are usually categorized under four main groups. The first group is based on the cluster formation methodology including top-down, bottom-up, and analytic optimization techniques [3]. A second group lists methods depending on the cluster model acquired such as hierarchical [4], centroid (as in K-means [5]), distribution such as expectation maximization [6], density [7, 8], subspace, group, and graph-based models [9, 10]. Thirdly, depending on the relationship type between entities and clusters, hard or soft clustering can be distinguished by defining binary or fuzzy relations, respectively. A final clustering group, based on the nature of cluster-cluster relations, defines the distinction between overlapping versus disjoint partition groups in general.

Challenges in Clustering Clustering problem is not a trivial task, especially in the case of high-dimensional data, found in most real-world applications. Conventional clustering methods usually fail in such scenarios. This phenomenon is referred to as *the curse of dimensionality* [11]. The problem here can be described with a synthetic example where there is a set of data originally in a low-dimensional space, which is gradually expanded with irrelevant information within some additional dimensions. As such dimensions are incrementally added, the

☆Accepted manuscript, DOI: 10.1016/j.sigpro.2018.02.010
Email addresses: Yigit.Oktar@ieu.edu.tr (Yigit Oktar), Mehmet.Turkan@ieu.edu.tr (Mehmet Turkan)
URL: http://people.ieu.edu.tr/en/yigitoktar (Yigit Oktar), http://people.ieu.edu.tr/en/mehmetturkan (Mehmet Turkan)
[1]Department of Computer Engineering, Izmir University of Economics, Izmir, Turkey. Corresponding author.
[2]Department of Electrical and Electronics Engineering, Izmir University of Economics, Izmir, Turkey.

inherent distribution of original data will gradually become obscure because of the increased volume, and that statistically sound subset becomes sparser in higher dimensions. This is especially problematic in the case of clustering, which employs some conventional distance metrics, as with each additional dimension, such functions will lose their discriminative power.

Large amounts of data does not mean that learning algorithms will be successful. *The problem of overfitting* [12] usually occurs when the model being captured is excessively complex because of very high-dimensional feature space. Also, the data at hand may not be a very representative of the whole ground-truth model. In that case, learning algorithms tend to fit a model to data samples at hand, thus missing the true underlying structure. In other words, some kind of memorization occurs instead of learning.

Noise and outliers are additional challenges to be tackled in practical signal processing and learning tasks. Especially, non-Gaussian noise is common in applications that involve measurements. Outliers, on the other hand, are inconsistent observations among the general population. Combined with certain output constraints, these peculiarities pose great challenges for problems involving both linear and non-linear systems. The effects of these additional considerations are best investigated in some recent studies as [13, 14, 15].

Remedies for Challenges Because raw data is usually in a crude form, as explained, clustering approaches require a preprocessing step to cope with high dimensions and undesired sampling issues.

Various preprocessing techniques have been proposed to increase preformance in cases of high dimensions, e.g., [16, 17, 18, 19]. They generally reshape the sample space through transformations or eliminations to observe the dataset in a refined way that would be more suitable for further processing. In an example, Principal Component Analysis (PCA) [19] is a method that transforms sample attributes into a form that would have the highest variance, thus more suitable for discrimination tasks with an additional benefit of reduced dimensions. In general, this concept can directly be generalized as *feature transformation*. However, it is important to note that such techniques do not discard irrelevant features, i.e., all features are preserved and reshaped through (non)linear combinations. As an extended approach, dimensionality reduction via feature selection performs elimination of irrelevant features and considers what seems to be the most important subset of features abiding by certain optimality criteria. Both of these techniques help future classification or clustering tasks to achieve more accurate representations in return.

Although feature selection can simply be used as a solution to high-dimensional problems, elimination process however might lead to some loss of important information that have strong meaning in different context, i.e., in different subspaces. In this light, *subspace search* [11], a combinatorial approach to subset selection, can be performed as an extension of feature selection where many subsets of features are distinctively analyzed while keeping original dimensions intact. A class of clustering methods has been proposed which includes searching for clusters in subspaces rather than the original space, thus referred to as *subspace clustering* [11]. Considering data points in isolated but relevant dimensions eliminates the interference of irrelevant dimensions, hence provides a solution to the clustering problem. Observing data in many alternate subspaces can provide means to clustering certain groups in certain subspaces, and the rest in different ones. The whole data can effectively be partitioned through merging the solutions in these subspaces, in a way that would not be possible by observing all dimensions at once. In general, subspace clustering problem can be formulated by defining the number of subspaces, subspace dimensions and a corresponding basis when supplied with a set of data points that lie within a union of subspaces.

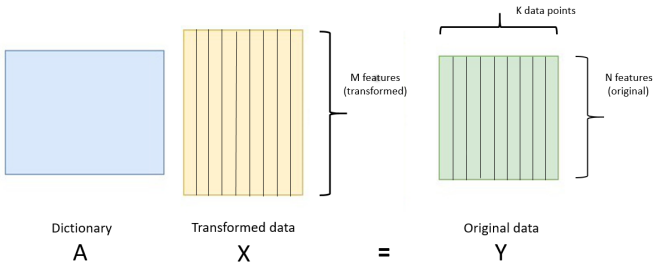

Figure 1: Sparse and redundant representations implement a transformation $\mathbf{Y} = \mathbf{AX}$ that increases the dimensionality of feature space from N to M and searches for suitable subspaces within the new M-dimensional feature space.

Existing subspace clustering approaches can roughly be categorized into four main groups [20]. K-subspaces [21] and Median K-flats [22] can be given as examples to iterative methods which alternate between assigning data points to subspaces and then fitting suitable cluster models to those assigned points. These approaches must usually be supplied with the number of subspaces to search for, and thus bound to initialization accuracy. Algebraic techniques include factorization-based [23, 24] and geometry-based approaches such as Generalized PCA (GPCA) [25]. Such methods provide improvements over previous ones, but are still susceptible to noise and outliers. There also exist various statistical approaches such as Mixtures of Probabilistic PCA (MPPCA) [26] and Random Sample Consensus (RANSAC) [27] with their advantages and disadvantages. Usually, such methods cannot provide a general solution for noise handling, and for defining the number of subspaces and subspace dimensions problems simultaneously. Finally, there are spectral clustering methods, which can be divided into two subcategories, namely local and global. Local spectral-clustering based approaches such as Local Subspace Affinity (LSA) [28] and Spectral Local Best-fit Flats (SLBF) [29] use local similarity measures on data points, but these are problematic in cases of subspace intersections and sensitive to parametric choice of neighborhood size. Global spectral-clustering based approaches instead use global information to find more suitable similarity measures between data points, e.g., [20].

Note that it is also possible to take a different approach to subspace search within expanded dimensions through *sparse and redundant representations* [30], which implements a transformation that increases the dimensionality of feature space, as

illustrated in Fig. 1, and then searches for suitable subspaces within this new feature space. This review paper focuses on the sparsity-based clustering methods, considering sparse and redundant representations as a both feature transformation and a "structure" of clustering with the help of adaptive (learned) overcomplete dictionaries.

Having considered the challenge of high dimensional data, it is also important to mention certain techniques to overcome the problem of overfitting. In the case of clustering, a common way to deal with overfitting is to minimize within cluster variance [31]. More generally, overfitting occurs when the model accommodates more parameters than needed [32]. In the sparse and redundant representations framework, the sparsity measure directly corresponds to parameter quantity, and can be manipulated easily. For example, by keeping the sparsity constraint strict enough, a model based on a few parameters can be formed, exhibiting less overfitting.

In the case of noise and outliers, it has been shown that under certain conditions, it is highly probable that sparse and redundant representations admit a local minimum around the reference signal-generating model [33]. This means sparse representations are indeed resilient to noise and outliers when certain criteria are met, such as appropriate scaling of dimensions, number of measurements, and model parameters. In practical applications, the nature of noise may not be Gaussian, but exhibit high levels of outliers. There are successful studies which overcome such situations with flexible structures for noise handling, e.g., a method based on a hybrid norm for minimizing the data fitting error term [34], a nonparametric scheme minimizing some norms of residual and original signals [35]. Furthermore, sparse and redundant representations are widely used in signal denoising applications, which provides the potential for robust models, even in the presence of (non-)Gaussian noise and outliers [36, 37, 38].

Considering all of the challenges, namely high dimensionality, high nonlinearity, parameter interactivity, and disturbances due to randomness at the same time, it may be more efficient to sample a set of solutions instead of searching the solution space completely. The complexity of practical real-world problems usually makes it almost impossible to fully search the solution space, and in such cases, metaheuristic approaches can be utilized as high-level procedures providing a sufficient solution to the optimization problem at hand [39]. There are many successful applications based on this approach, e.g., [40, 41, 42]. More specifically, it is possible that, computationally-hard sparse and redundant representations can be attained through various metaheuristic methods as presented in recent studies [43, 44].

The rest of this paper is organized as follows. Section 2 first overviews the problem of sparse representations, and then relates the clustering problem to the sparsity constraint, namely, to *sparse coding*. Following this, Sec. 3 introduces the principles of *dictionary learning* for sparse representations, and then connects the clustering problem to learned dictionaries. Section 4 then discusses related concepts to sparsity-based clustering, and describes open problems while plotting possible future directions in this domain of research. Finally, Sec. 5 draws a brief conclusion.

2.Sparse Representations and Clustering

2.1.Sparse representations: An overview

Sparse representations have become a key research topic with various applications in signal and data processing, e.g., denoising [36, 45], modeling [46], restoration [47, 48], compression [49, 50], and even more [51, 52, 53]. Put simply, sparse representations represent most or all information contained in a data with a weighted linear combination of a small number of elements or *atoms* chosen from an overcomplete or redundant basis or *dictionary*. Such a dictionary is a set of atoms whose number is much larger than the dimension of the data space. Any entity then admits an infinite number of representations, and the sparsest such solution has interesting aspects for various data processing tasks.

The main objective is to obtain a sparse approximation of a given input data $\mathbf{y} \in \mathbb{R}^N$. Given a full rank matrix $\mathbf{A} \in \mathbb{R}^{N\times M}$ with $N \leqslant M$, one tries to optimize the solution of

$$\mathbf{y} = \mathbf{A}\mathbf{x} \;\text{ subject to }\; \min \|\mathbf{x}\|_0 \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^M$ denotes the sparse representation of $\mathbf{y}$ and $\|\mathbf{x}\|_0$ is the ℓ_0–norm of $\mathbf{x}$, i.e., the number of non-zero components in $\mathbf{x}$. The matrix $\mathbf{A}$ is the *dictionary* and its columns (*atoms*) are assumed to be normalized in any norm.

In general, sparse representations for any set of data can be imposed in the form of a matrix factorization as $\mathbf{Y} = \mathbf{A}\mathbf{X}$, where $\mathbf{Y}$ denoting the original data with N features and K samples, and $\mathbf{X}$ as the sparse representation matrix of $\mathbf{Y}$ in the new M-dimensional feature space as depicted in Fig. 1. Assuming that $\mathbf{A}$ is fixed, the ℓ_0–norm constraint on the columns of $\mathbf{X}$ forces each data sample to use only a small number of feature templates (atoms). Hence, sparse codes, namely the columns of $\mathbf{X}$, together with the atoms they use, define a subspace.

In practice, the whole problem can be relaxed as an approximate convex optimization while fixing $\mathbf{A}$ and solving for $\mathbf{x}_i$ for each $\mathbf{y}_i$ $\forall i$ independently, by minimizing the total approximation error over all samples by

$$\arg\min_{\{\mathbf{x}_i\}} \sum_{i=1}^{K} \|\mathbf{y}_i - \mathbf{A}\mathbf{x}_i\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}_i\|_0 \leq k \;\forall i, \tag{2}$$

which is known as *the sparse coding problem*. Here, the parameter k defines the maximum sparsity allowed for the representation of $\mathbf{y}_i$ during the sparse coding process.

There is no known technique for obtaining the exact solution under general conditions on the fixed dictionary $\mathbf{A}$, except for the exhaustive combinatorial approach. Searching for this sparsest representation is hence unfeasible. This problem is computationally intractable [54] and thought to be NP-hard [55]. A wide variety of pursuit algorithms [56, 57, 58, 59] have been introduced as heuristic greedy methods aiming at approximate solutions with tractable complexity.

For the ℓ_0–norm constraint, greedy approaches are the most appropriate as the above problem in this form is NP-hard. Matching Pursuit (MP) [57] and Orthogonal MP (OMP) [58] are most widely used examples to these iterative methods.

On the other hand, it has been shown that for many high-dimensional cases, ℓ_1–norm constraint (instead of ℓ_0–norm) is sufficient to ensure the sparsest solution [60]. Note that the very same problem with ℓ_1–norm constraint can then be solved via regular linear programming tools, such as interior point [61] or regression shrinkage [62]. Basis Pursuit (BP) [56] is the generalized term for ℓ_1–norm constrained version, as an approximation to the original problem.

Note that, since the transformed feature space in $\mathbf{X}$ has higher dimensions than that in $\mathbf{Y}$, this feature transform can also be coined as "dimensionality expansion", as opposed to dimensionality reduction such as in PCA. This is an advantage because, through dimensionality expansion, it is possible to utilize the subspace clustering approach at large.

2.2.Sparse coding linked with clustering

Sparse coding can be thought of as a method of information localization. In this sense, sparse representations and the clustering problems are usually complementary, as clustering itself includes a form of information localization. However, note that the sparsity constraint alone does not imply clustering. For example, random sparsity is an expression of information being localized to a certain extent, but clustering would not be evident at all for such a case. Thus, sparsity property needs indeed to be structured in order to be significant in informative sense. By nature, in most real world examples sparsity property and clusters are usually observable together. As an example, in social networks, not everyone is friends with everyone else, and people appear to be in certain friendship groups. Similarly protein-protein interactions in molecular biology are selective, while proteins of same functional domain and cellular location tend to cluster [63]. A striking sparsity example can be given for the brain. The brain, that fundamentally based on compartmentalization, not only has spatial but also temporal sparsity. Neurons are active in relatively small number of time periods, and also, the activated population of neurons are spatially sparse, i.e., only a small portion of neurons are active at any time [64].

At this stage, there are two possible directions towards a solution for the main topic, namely for the clustering problem. Firstly, it is possible to supply sparse representations (i.e., sparse codes) acquired to any existing data clustering method –as extracted features– to be further processed as exemplified in [20]. In this simple case, sparse representations remains as a tool of feature transformation and/or selection, as a preprocessing step for clustering. Secondly, it is possible to formulate the sparsity concept as a clustering problem directly through additional structural constraints on sparse and redundant representations.

2.2.1.Sparsity as a feature transform

The first option is to use the transformed feature space, namely sparse codes, as an input to any existing data clustering algorithm. As a special case, if the dictionary is chosen as the data itself, i.e., $\mathbf{A} = \mathbf{Y}$, the result is a formulation as $\mathbf{YX} = \mathbf{Y}$. In this form, $\mathbf{X}$ contains information about a kind of *self-similarities* among the original data. However, the diagonal entries of $\mathbf{X}$ has to be forced to be zero to prevent the trivial solution of $\mathbf{YI} = \mathbf{Y}$, where $\mathbf{I}$ represents the identity matrix with suitable dimensions. The columns of the dictionary are usually normalized, arriving at a final formulation as $\hat{\mathbf{Y}}\mathbf{X} = \mathbf{Y}$ where $\hat{\mathbf{Y}}$ denotes $\mathbf{Y}$ with normalized columns. In an example, this logic is utilized to solve the problem of segmenting multiple motions in videos [65, 20]. After solving for $\mathbf{X}$ through ℓ_1–norm constraint optimization, a similarity matrix is further calculated by $|\mathbf{X}| + |\mathbf{X}^T|$, which is then processed by spectral clustering for final segmentation. Experiments on chosen video sequences show that this approach is exceptionally successful in this clustering task.

If above $\hat{\mathbf{Y}}\mathbf{X} = \mathbf{Y}$ is solved for $\mathbf{X}$ with a greedy approach, such as MP or OMP with an ℓ_0–norm constraint where $k = 1$, then the non-zero coefficient with the index j within $\mathbf{x}_i$, will show that $\mathbf{y}_i$ and $\mathbf{y}_j$ are the most similar in terms of directionality; in other words $\mathbf{y}_i$ and $\mathbf{y}_j$ are highly correlated in terms of angular similarity. This can be regarded as an alternative similarity measure to Euclidean distance. Note also that for any sparsity constraint with $k > 1$, this formulation can be generalized as directional decomposition of the data.

As a side note, there is in fact a whole field of directional statistics [66, 67], in which data points are represented as scaled directions –contrary to points in cartesian coordinates– and their distributions are examined from that perspective. In that regard, Von Mises-Fisher probability formulation deals with distributions on circles, spheres, or in general n-dimensional hyperspheres [68]. In relation to the clustering problem, cosine similarity or angular distance as its inverse, can be used alternatively. Spherical K-means [69] aims to maximize the cosine similarity objective, thus it is equivalent to K-means clustering on a unit hypersphere. Von Mises-Fisher directional distributions can also be used as a probabilistic approach for clustering [70, 71].

2.2.2.Structured sparsity as a means of clustering

Instead of using sparse codes as features in the existing clustering algorithms, sparse code appearance patterns can directly be utilized for clustering. While noting that there is no structural constraint on code-patterns in the conventional sparse coding approaches, additional structural code-pattern constraints can easily be injected and then manipulated so that subspaces can directly be designated as clusters. As a simple example, Fig. 2 depicts a coding case where, as well as two disjoint subspaces as desired (green and blue code-patterns), there also exists non-structural patterns (green and yellow, blue and yellow). Structural constraints can enforce the condition that all sparse code-patterns appear in disjoint subspaces, which will naturally designate structured clusters.

As a structured sparsity technique, *group sparsity* enforces grouping of the elements belonging to the sparse-code vectors by allowing coefficients to fill the vector group-by-group. The sparse code is conceptually partitioned into overlapping or disjoint groups, and an additional norm constraint is used on this group level. In this kind of structure, there is a cascade of norm constraints, usually two-layered, as opposed to a single, general one in the conventional sparse coding approaches. As an example, ℓ_1–norm constraint based well-known *lasso* [62] method

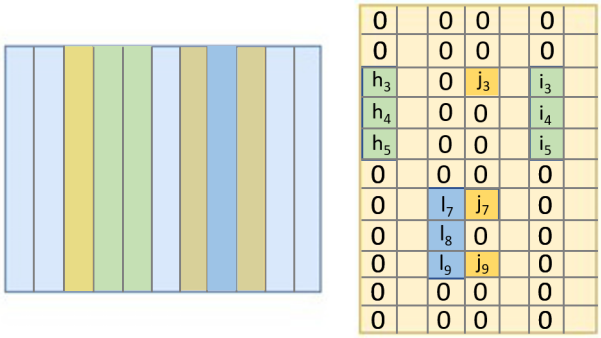

Figure 2: An example sparse coding case: (left) a portion of the dictionary is shown with colored columns (atoms), and (right) example sparse code-patterns colored with the same color as the atoms used during coding. There are two disjoint subspaces (green and blue code-patterns), and there also exists non-structural code-patterns (green-yellow and blue-yellow).

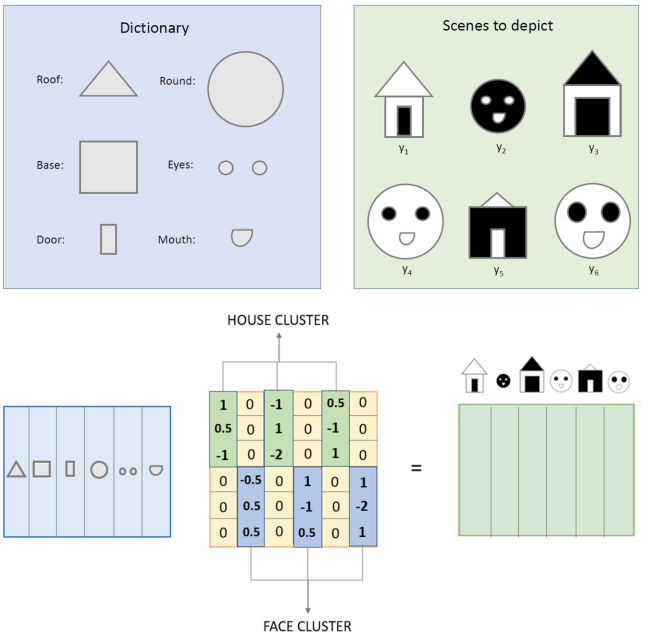

Figure 3: A block sparsity example illustrating the designation of disjoint subspaces which are assigned as clusters: house (green) and face (blue) clusters. The dictionary and the data represented in the sparse domain are shown on the top-row.

can be extended to *group lasso* [72] with an ℓ_2–norm constraint on the group level.

Considering the sparsity concept as a cascade of norm constraints forced on sparse codes on multiple levels leads to the possibility of multiple norm combinations, and also to other structural variations. An example is *sparse group lasso* [73], which extends group lasso through a global ℓ_1–norm constraint in addition to ℓ_1–norm group sparsity and ℓ_2–norm within group constraints. Such an enforcement yields sparsity both on the group and global levels, whereas group lasso alone does not enforce sparsity within a group. As a second example, *strong group sparsity* [74] has ℓ_1–norm within group constraint, and the support selected is restricted to that lying within the smallest possible subset of non-overlapping groups. Through a generalization, any structure can further be imposed on the sparse code set. In a recent study [75], group sparsity was extended through a subset imposing cost function while defining the coding complexity for that sparse subset. Note that by manipulating such cost functions, a range, including block, hierarchical or even graph sparse code-patterns can be enforced. It is important to keep in mind that structured sparsity is a sparse coding approach that will work particularly well if the data itself has that specific structural nature [74]. Without initial structure, the structured coding will much be less meaningful.

Most relevant to the clustering problem, *block sparsity* [76, 65] is a specific case of disjoint group sparsity where groups appear in blocks. A block sparsity of level 1 corresponds to some designation of disjoint subspaces. In such case, these subspaces can directly be assigned as clusters and that will correspond to a non-overlapping subspace clustering scheme as depicted in Fig. 3. Going further, just by considering dictionary atoms as directions, sparse codes in these subspaces themselves can be regarded as directional signals, and such a perspective paves way to a hierarchical clustering scheme that can even descend to the magnitude level. While investigating the sparse codes of the example in Fig. 3, it is seen that the pair $[1, -1, 0.5]$ and $[1, -2, 1]$ lie in the same octant as the signs of coordinates match. Thus, sparse codes within subspaces can further be categorized under specific *orthants* (in n-dimensions), depending on their coordinate signs. Furthermore, a pair that matches in direction (ignoring the scale), such as $[1, -2, 1]$ and $[2, -4, 2]$, will constitute a directional categorization below the orthant level. Finally, to distinguish within the same direction, sparse codes can further be analyzed by their scale or magnitude. This scheme is illustrated in Fig. 4. Note that, at the very bottom level, ℓ_1–norm discrimination is equivalent to that of ℓ_2–norm, i.e., $|x| = \sqrt{x^2}$.

3.Dictionary Learning and Clustering

3.1.Dictionary learning: An overview

A crucial question in sparse representations is the choice of the dictionary. Possible choices include various sets of analytic waveforms such as overcomplete DFT, DCT, wavelets. However, both the sparsity and the quality of the representation depend on the used dictionary, and most importantly its suitability for the data and the problem at hand. Therefore, the underlying main idea of dictionary learning for sparse representations suggests that the data can be better approximated sparsely as a weighted linear combination of a set of *prelearned* dictionary atoms, rather than off-the-shelf overcomplete bases or dictionaries, e.g., [36, 45, 46, 47, 48, 49, 51]. The sparsity constraint associated with the learning problem generally leads to a solution which can fit any practical application by means of pursuit algorithms with ℓ_0–norm and ℓ_1–norm sparsity measures.

The objective is to obtain an explicit dictionary matrix $\mathbf{A}$ which is optimally representative of a given set of training samples under some strict sparsity constraints. Formally, given a set of training data with N features and J samples stored in the

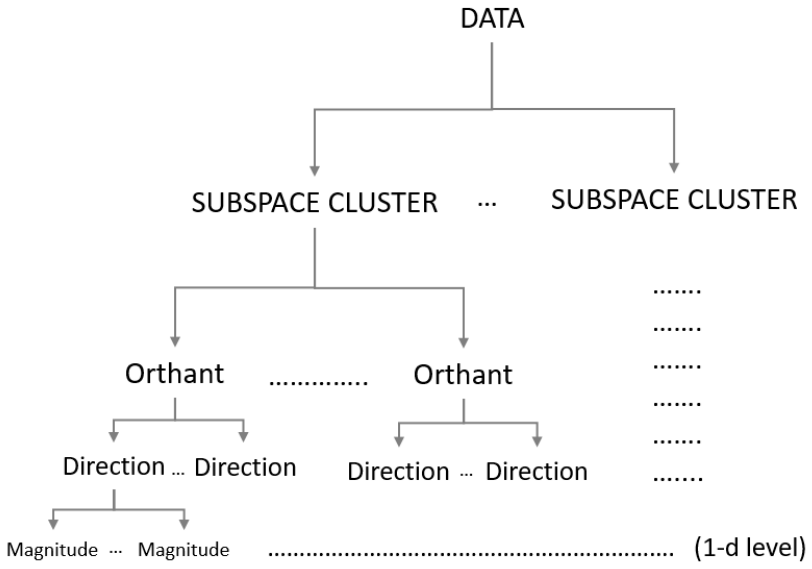

Figure 4: A perspective of sparsity-based hierarchical clustering that descends to the magnitude level.

columns of a matrix $\mathbf{T}$, the search for an optimum dictionary $\mathbf{A}$ involves solving the constrained minimization as

$$\arg\min_{\mathbf{A},\mathbf{Z}} \|\mathbf{T} - \mathbf{A}\mathbf{Z}\|_F^2 \quad \text{subject to} \quad \left\|\mathbf{z}_j\right\|_0 \leq k \;\forall\, j \tag{3}$$

where the sparse matrix $\mathbf{Z} \in \mathbb{R}^{M\times J}$ has its columns $\mathbf{z}_j$ as the sparse representation vectors of the corresponding training samples $\mathbf{t}_j \;\forall\, j$. Note that the constraint on $\mathbf{A}$ is implicitly assumed to be valid to obtain unit norm atoms. Here $\|\,.\,\|_F$ denotes the Frobenius norm.

The problem in Eqn. 3 is combinatorial and highly non-convex, and thus a local minimum can be expected [77]. Alternatively, this formulation can be rewritten as a joint optimization with respect to the dictionary $\mathbf{A}$ and sparse vectors $\mathbf{z}_j \;\forall\, j$ while including the sparsity constraint in the formula as a penalty term

$$\arg\min_{\mathbf{A}} \sum_{j=1}^{J} \left\{ \arg\min_{\mathbf{z}_j} \left[\left\|\mathbf{t}_j - \mathbf{A}\mathbf{z}_j\right\|_2^2 + \alpha_j \left\|\mathbf{z}_j\right\|_0 \right] \right\} \tag{4}$$

which is not jointly convex but convex with respect to one of its variables when the other one is fixed [36]. α_j here represents the sparsity regularization parameter for $\mathbf{t}_j \;\forall\, j$.

In this way, the whole problem can be factorized into two *approximate* convex optimization steps as: a) *sparse coding*: optimizing $\mathbf{z}_j \;\forall\, j$ by fixing $\mathbf{A}$; b) *dictionary update*: optimizing $\mathbf{A}$ by fixing $\mathbf{z}_j \;\forall\, j$. A solution can be reached by iteratively solving these two steps.

While sparse codes $\mathbf{z}_j \;\forall\, j$ can be calculated in the sparse coding step as discussed in Sec. 2, the problem is then to optimize $\mathbf{A}$ by minimizing the representation error of the training samples. This optimization is known as *the dictionary update problem*, and it can be formulated as

$$\arg\min_{\mathbf{A}} \sum_{j=1}^{J} \left\|\mathbf{t}_j - \mathbf{A}\mathbf{z}_j\right\|_2^2. \tag{5}$$

The above described optimization problem can be solved using various techniques. Non-parametric dictionary learning methods, such as Method of Optimal Directions (MOD) [78] and K–SVD [79], have been developed, resulting in non-structural learned dictionaries. There are also parametric learning structures for such as *translation invariant dictionaries* [80, 81, 82, 83], *multiscale dictionaries* [48, 84], *unions of orthonormal bases* [85, 86] and *sparse dictionaries* [87]. Moreover, for various data and signal processing tasks, the literature provides online learning algorithms [88, 89], task-driven learning approaches [90], tree-structured hierarchical methods [91, 92, 93], and iteration-tuned schemes [94, 95].

3.2.Dictionary learning linked with clustering

As mentioned above, a chosen fixed dictionary may not always be appropriate for clustering the data at hand, especially when the data under investigation is of an unknown nature. This situation is exemplified in Fig. 5, which builds on top of the previously selected example, depicting that sparse codes are not structured, but lie apparently on intersecting subspaces. There is no obvious cluster-like appearance in sparse codes. However, it is possible to acquire disjoint subspaces through adapting the dictionary to the data at hand, by learning a more suitable dictionary.

Constraining block sparsity structure onto sparse codes using the sparse coding step, followed by a dictionary learning step, basically corresponds to learning a block-sparse model for the data. This is a form of non-overlapping clustering, where distinct subspaces are defined by adaptive subdictionaries together with the supporting block of sparse codes. Examples from literature prove that such an approach is a successful alternative to self-similarity. As an example in [96], subdictionaries are modeled to learn a subspace for each cluster in an image segmentation task. Initially, data samples are assigned to best representing subdictionary according to a certain representation

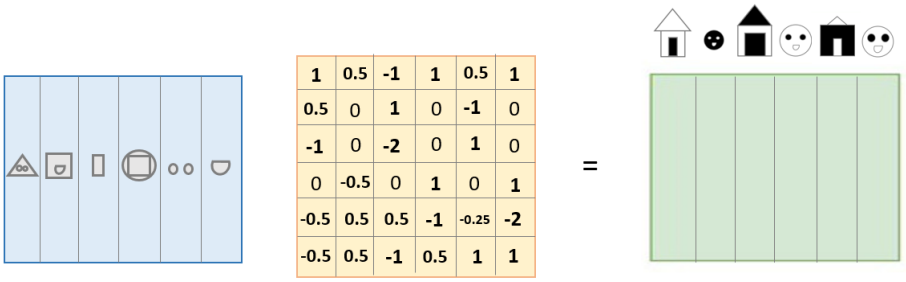

Figure 5: The chosen dictionary may not always be appropriate for clustering the data. An example sparse decomposition depicts that sparse codes are not structured, but lie on intersecting subspaces.

quality that includes a data fidelity term and a sparsity promoting term. Then, these assignments are fixed, and solutions for better adapted subdictionaries are calculated with an additional incoherence term. This proves to be an efficient and effective solution for the image segmentation problem targeted. A successful denoising application example can also be given in [97]. Here, the concept of *double-header* ℓ_1*–optimization* is introduced with an additional ℓ_1–norm restriction that enforces best representing centroid for each cluster through an adaptive dictionary. Simultaneous centroid enforcement and sparse coding create a noise-resilient structure. Encouragingly, this denoising application is reported to match the state-of-the-art BM3D [98] performance. In fact, denoising is a very suitable domain for such clustering formulations. Yet in another denoising study in [99], authors successfully aim at the decomposition of images into multiple semantic layers through unsupervised clustering based on self-learning, allowing detection and removal of undesired patterns such as Gaussian noise and rain strikes.

4.Related Concepts and Open Problems

4.1.Signal processing perspective

4.1.1.Compressive sensing

Compressive sensing (CS) aims at reducing the number of measurements needed to describe a signal while exploiting its compressibility. It can mathematically be expressed by

$$\mathbf{z} = \mathbf{\Phi y} = \mathbf{\Phi \Psi x} = \mathbf{\Theta x} \tag{6}$$

where $\mathbf{\Phi} \in \mathbb{R}^{N \times M}$ is *the stable measurement matrix* with $N < M$, and it is responsible of dimensionality reduction from $\mathbf{y} \in \mathbb{R}^M$ to $\mathbf{z} \in \mathbb{R}^N$ such that $\mathbf{z}$ designates less number of measurements taken. The main goal is to recover the original compressible signal $\mathbf{y}$, or equivalently the sparse signal $\mathbf{x}$, from $\mathbf{z}$. Note here that $\mathbf{\Psi} \in \mathbb{R}^{M \times M}$ is an orthonormal sparsifying basis for obtaining k-sparse representation signal $\mathbf{x}$ such that $\mathbf{y} = \mathbf{\Psi x}$ and $\mathbf{x} = \mathbf{\Psi}^T \mathbf{y}$. k largest coefficients in $\mathbf{x}$ are kept while discarding the smallest for $k \ll M$ [100]. The solution to this problem involves two steps. First, a suitable $\mathbf{\Phi}$ has to be designed, and then a reconstruction algorithm is needed to recover $\mathbf{y}$ from $\mathbf{z}$. For stability, a sufficient condition is that $\mathbf{\Theta} = \mathbf{\Phi \Psi}$ satisfies the restricted isometry property (RIP) [101]. An alternative approach for stability is to ensure that the matrix $\mathbf{\Phi}$ is incoherent with the sparsifying basis $\mathbf{\Psi}$. However, in practice, the signal $\mathbf{y}$ at hand may not be sufficiently sparse in an orthonormal basis, but in a redundant and overcomplete dictionary. Through a generalization, $\mathbf{\Psi}$ then can be replaced with a highly overcomplete and coherent dictionary $\mathbf{A}$ tying the gap between CS and sparse representations [102].

Considering compressible signals in a "*structured nature*" paves way to model-based CS. These methods significantly decrease the bound of required measurements to $N = O(k)$ for tree-sparse and (in the limit) for block-sparse signals, whereas standard CS models can robustly recover k-sparse compressible signals from $N = O(k \log(M/k))$ measurements. Such approaches to CS have helped to decrease the required amount of measurements for robust recovery of signals in applicable domains. Similar methods can also be used for CS recovery of clustered signals [103, 104, 105].

4.1.2.Multiple measurement vectors

Up to this point, sparse and redundant representations have been considered through a single measurement vector (SMV) framework, in which each signal is considered individually, even though sparse representations solution is obtained for multiple signals. In the multiple measurement vectors (MMV) approach, on the other hand, multiple signals are simultaneously considered by processing multiple sparse vectors together while selecting a column (an atom) from the dictionary $\mathbf{A}$. The optimization of MMV can be formulated as

$$\arg\min_{\mathbf{X}} R(\mathbf{X}) \;\text{ subject to }\; \mathbf{Y} = \mathbf{AX} \tag{7}$$

where $R(\mathbf{X})$ represents the number of rows in $\mathbf{X}$ containing non-zero entries [106, 107].

The perspective of MMV is especially powerful when solutions have an initial common sparsity profile. However, dictionary learning with MMV approach will be extremely ill-posed because all-zero rows in $\mathbf{X}$ may cause corresponding dictionary atoms either to disappear or to diverge during the dictionary update step. However, this approach can be successfully used to discard certain atoms from a highly overcomplete dictionary to obtain a more compact representation. As a final relevant note, algorithms similar to ones used for MMV recovery can be adapted for the block sparsity structure, thus can be linked to clustering [108, 109, 110, 76].

4.2.Machine learning perspective

4.2.1.Large margin modeling, clustering and sparsity

Large margin modeling can be defined as finding hyperplanes which maximize the margin between classes [111, 112, 113]. Such a model is composed of a bias, weight vectors and support vectors. Support vectors are selected as data points which lie closest to hyperplanes, and these are sufficient to express the whole data set. In other words, support vectors lie on the margin and, for certain applications, carry all the relevant information about the data. Thus, the solution is sparse in nature. It has been shown that a slight modification of ℓ_1–norm sparsity optimization method, namely Basis Pursuit, is equivalent to Support Vector Machines (SVMs) [114], which are large margin formulations [115]. Building on top, large margin clustering is also possible through maximizing inter-cluster margins [116, 117]. Therefore, drawing parallels between the two variants of this approach can lead to a more generalized theory that is able to capture the gist of the sparsity concept analytically, since both variants can be thought as mathematically constructive methods.

4.2.2.Symbolic abstraction

Symbolic machine learning is traditionally associated with ID3 decision tree learning [118]. In general, the symbolic approach to machine learning can be thought as inductive learning, in which certain rules are inversely deduced from the observed data. However, symbolism has a broader presence in the AI world in general, as a means of high-level abstraction over numerical units, often introducing human-readable representations [119]. In line with this definition, model-based clustering can be classified as an approach to symbolism. For example, centroids in K-means, as rather shallow symbols, with a distance rule for assignments, define an abstract object that provides partitioning. Similarly, large margin modeling can be seen as another shallow symbolic approach; in this case, support vectors with their strict boundaries provide an abstraction layer. In this sense, symbols, as abstracted objects, provided with rules for decisions can be regarded as sparse representations, since this approach allows a relatively small number of symbols to express enormous amounts of data. It is important to note that the shallow analogies given here may not be common, as symbols generally need to be very high-level abstractions, as in [120, 121].

4.2.3.Artificial neural networks and deep learning

Connectivist approaches, such as traditional neural networks [122], tend to process data in pure numerical units. Perceptron is a generalization of a single neuron cell that works on the numerical unit level [123]. However, with only a single layer, perceptrons are not capable of learning the nonlinearity. A multi-layer generalization of perceptrons solves this problem, while also introducing a possibility for sparsity through the activation function [124]. However, such generalization still depends on numerical units for computation. Convolutional Neural Networks (CNNs) provide an abstraction over multi-layer structure, in which a degree of "symbolism" is introduced, as apparent from the human-readable filters that are formed within the nodes [125]. Note that connectivism, rather than enabling deep understanding, simply replicates the evolved structure of the human brain, unlike analytic approaches. As a recently popularized approach, clustering with deep learning [126, 127] at this stage may be successful, but currently it is not sufficient to provide a deep analytic understanding of inner-working procedures of sparse structures and clustering peculiarities.

4.2.4.Deep sparse structures

As a more refined version of hierarchical clustering illustrated in Fig. 4, block sparsity structure can recursively be enforced onto sparse codes, and therefore subdictionaries can be learned on different levels in a *deep* manner. After acquiring block-sparse codes in the first level, one can appropriately rearrange these sparse codes in a block-diagonal form, as depicted in Fig. 6, allowing a new block sparsity decomposition to be performed on successive levels. Such a system will result in a deep block-diagonal decomposition of data, which resembles a taxonomy. In fact, this decomposition divides the data into disjoint clusters at each level. This structure generally resembles a decision tree structure [118]. However, it can also be utilized in an unsupervised way, unlike decision trees which usually need labeling.

4.2.5.A unifying framework

There are five general schools in machine learning, as described in [128]. *Symbolists* employ inverse deduction in order to construct logical rules and abstractions. *Connectionists* try to reverse-engineer the brain functioning. *Evolutionists* simulate the process of biological evolution. *Bayesians* use statistics to pose learning as a form of probabilistic inference. *Analogizers* mainly use mathematical formulations and optimization tools. Then, according to the text, a prospective *master algorithm* for general learning needs to incorporate the essence of these five schools.

The proposed deep sparse structure inherently incorporates symbolic and analytic approaches. Block-sparse coding is an analytic method, whereas dictionary update with block-sparse codes corresponds to a form of inverse deduction where recursive subdictionaries represent abstract objects with their rules of division at each level, and this procedure can be recognized as a symbolic approach. Moreover, the deep structure includes connectivism, but since this is only a forward learning procedure, it needs an additional tool analogous to back-propagation. This can be realized through the multi-level generalization of error-feedback procedure, which has previously been proposed in [129].

A key challenge of clustering that has not been extensively touched upon in this review is determining the inherent number of clusters within the data, and their dimensions. As is, the proposed scheme requires an exhaustive search for determining the optimal number of blocks and their dimensions at each level, which is clearly infeasible. However, an evolutionary approach would allow substructurally different dictionaries compete for approximation accuracy (the fitness function) through inter-breeding, as a heuristic solution.

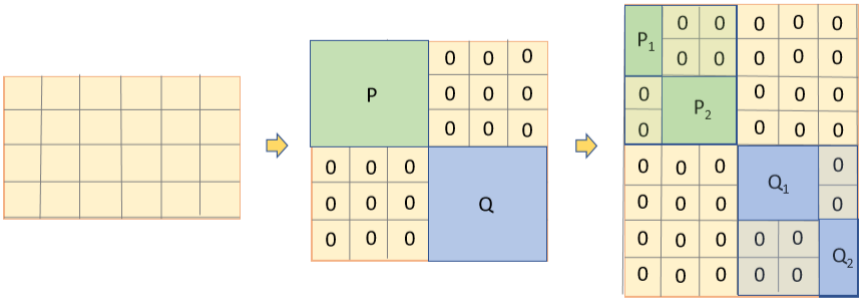

Figure 6: Deep sparse structures: deep block-diagonal decomposition of data.

It is important to note here that clustering as an unsupervised way of learning may not represent the ground truth as is. Thus, as a final step, Bayesian statistics can be utilized to map the cluster entities to actual classes that are known a priori, closing the gap between the observed and the known.

4.3.Required parameters and initial conditions

Although many different design variations of the proposed framework are possible, initial parameter quantities and preconditions must be specified to begin with. First of all, it is important to determine the size of the dictionary $\mathbf{A}$ together with the sizes of each subdictionary. The number of (sub)dictionary rows is predetermined by the dimension of the signal space, and the number of columns of each subdictionary determines the dimensions of the corresponding subspaces. In block sparsity of level 1 formulations, the subdictionary sizes directly correspond to sparsity of each supposed cluster, therefore sparsity levels need not be further considered as parameters. Additionally, there is a need to define the number of clusters to search for, namely the number of subdictionaries, accomplished by certain procedures [130]. In short, the number of clusters to search for and their respective subspace dimensions are the required parameter quantities. Certain adaptations as mentioned in Sec. 4.2.5 may solve this problem by successfully recovering both the number of clusters and their respective dimensions. In this case, the only quantity that has to be provided is an initial dictionary of arbitrary size with arbitrary substructure.

Dictionary initialization is perhaps the most important precondition that affects the outcome, in addition to the parameters defined above. Converged subdictionaries correspond to bases for each subspace, to determine the clustering model that is searched. Therefore, the initial dictionary is a crucial factor for the convergence characteristics of the clustering model. Note here that for block sparsity, initialization of the dictionary should be handled as a special case due to the existence of several subdictionaries. A fully random initialization may be problematic, due to the similar initial nature of each subdictionary, hence a procedure that makes a clear distinction between subdictionaries should be selected. A preclustering scheme may be deployed to initialize subdictionaries, but in this case, the converged state will be highly dependent on this initial clustering. A more effective approach would be to precluster randomly generated atoms into respective subdictionaries, or else certain incoherent off-the-shelf subdictionaries can be selected.

5.Conclusion

In this paper, a review of sparsity-based clustering methods has been presented. Sparsity-based clustering, as a form of subspace clustering, can be regarded as a subspace designation tool within expanded dimensions. Dimension expansion through the usage of an overcomplete dictionary and coexisting structural sparsity constraints, such as block sparsity, create conditions for a more comprehensive subspace clustering scheme. The dual of the sparsity concept, namely dictionary learning, with coexisting constraints, is yet another aspect that can be utilized to adapt state-model entities into well-formed clusters.

As noted, usage of sparse and redundant representations in clustering has many parallels in various signal processing and machine learning tasks. In signal processing domain, relevant block-sparse formulations are considered within compressive sensing and multiple measurement vectors domains. Various machine learning approaches are more comprehensively linked to sparsity concept, providing an outline for a promising, unifying deep block-sparse model for hierarchical clustering. As a final concluding remark, sparsity concept is an influential notion, and its rigorous investigation may lead to discoveries affecting various scientific fields.

References

[1] G. Borgefors, Distance transformations in digital images, Comp. Vis. Graph. Image Process. 34 (3) (1986) 344–371.
[2] R. D. Maesschalck, D. J. Rimbaud, D. L. Massart, The mahalanobis distance, Chemometrics Intelligent Lab. Syst. 50 (1) (2000) 1–18.
[3] A. D. Gordon, A review of hierarchical classification, J. R. Stat. Soc. Series A (1987) 119–137.
[4] R. Sibson, SLINK: An optimally efficient algorithm for the single-link cluster method, The Comp. J. 16 (1) (1973) 30–34.
[5] S. Lloyd, Least squares quantization in PCM, IEEE Trans. Inf. Theory 28 (2) (1982) 129–137.
[6] C. Carson, S. Belongie, H. Greenspan, J. Malik, Blobworld: Image segmentation using expectation-maximization and its application to image querying, IEEE Trans. Pattern Anal. Mach. Intell. 24 (8) (2002) 1026–1038.
[7] M. Ester, H. P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proc. Knowledge Disc. Data Mining, Vol. 96, 1996, pp. 226–231.
[8] H. P. Kriegel, P. Kroger, J. Sander, A. Zimek, Density-based clustering, Data Mining and Knowledge Discovery 1 (3) (2011) 231–240.
[9] P. F. Felzenszwalb, D. P. Huttenlocher, Efficient graph-based image segmentation, Int. J. Comp. Vis. 59 (2) (2004) 167–181.

[10] P. Novak, P. Neumann, J. Macas, Graph-based clustering and characterization of repetitive sequences in next-generation sequencing data, BMC Bioinformatics 11 (1) (2010) 378.
[11] L. Parsons, E. Haque, H. Liu, Subspace clustering for high dimensional data: a review, ACM Sigkdd Explorations 6 (1) (2004) 90–105.
[12] D. M. Hawkins, The problem of overfitting, J. Chem. Inf. Comput. Sci. 44 (1) (2004) 1–12.
[13] V. Stojanovic, N. Nedic, D. Prsic, L. Dubonjic, Optimal experiment design for identification of ARX models with constrained output in non-Gaussian noise, Elsevier App. Mathematical Modell. 40 (13) (2016) 6676–6689.
[14] V. Stojanovic, V. Filipovic, Adaptive input design for identification of output error model with constrained output, Circuits, Syst., Signal Process. 33 (1) (2014) 97–113.
[15] V. Stojanovic, N. Nedic, Identification of time-varying OE models in presence of non-Gaussian noise: Application to pneumatic servo drives, Int. J. Robust and Nonlinear Control 26 (18) (2016) 3974–3995.
[16] G. Hinton, R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[17] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: A general framework for dimensionality reduction, IEEE Trans. Pattern Anal. Mach. Intell. 29 (1) (2007) 40–51.
[18] G. Baudat, F. Anouar, Generalized discriminant analysis using a kernel approach, Neural Comput. 12 (10) (2000) 2385–2404.
[19] I. Jolliffe, Principal component analysis, Wiley Online Lib., 2002.
[20] E. Elhamifar, R. Vidal, Sparse subspace clustering: Algorithm, theory, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2765–2781.
[21] P. Tseng, Nearest q-flat to m points, J. Optimization Theory App. 105 (1) (2000) 249–252.
[22] T. Zhang, A. Szlam, G. Lerman, Median k-flats for hybrid linear modeling with many outliers, in: Proc. IEEE Comp. Vis. W., 2009, pp. 234–241.
[23] T. E. Boult, L. G. Brown, Factorization-based segmentation of motions, in: Proc. IEEE W. Vis. Motion, 1991, pp. 179–186.
[24] W. Xu, X. Liu, Y. Gong, Document clustering based on non-negative matrix factorization, in: Proc. ACM SIGIR Conf. Res. Dev. Inf. Ret., 2003, pp. 267–273.
[25] R. Vidal, Y. Ma, S. Sastry, Generalized principal component analysis (GPCA), IEEE Trans. Pattern Anal. Mach. Intell. 27 (12) (2005) 1945–1959.
[26] M. E. Tipping, C. M. Bishop, Mixtures of probabilistic principal component analyzers, Neural Comput. 11 (2) (1999) 443–482.
[27] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications ACM 24 (6) (1981) 381–395.
[28] J. Yan, M. Pollefeys, A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate, in: Proc. European Conf. Comp. Vis., 2006, pp. 94–106.
[29] T. Zhang, A. Szlam, Y. Wang, G. Lerman, Hybrid linear modeling via local best-fit flats, Int. J. Comp. Vis. 100 (3) (2012) 217–240.
[30] M. Elad, Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing, Springer, 2010.
[31] A. Demiriz, K. P. Bennett, M. J. Embrechts, Semi-supervised clustering using genetic algorithms, Artificial Neural Netw. Eng. (1999) 809–814.
[32] B. S. Everitt, A. Skrondal, The Cambridge dictionary of statistics, Cambridge University Press, 2002.
[33] R. Gribonval, R. Jenatton, F. Bach, Sparse and spurious: Dictionary learning with noise and outliers, IEEE Trans. Inf. Theory 61 (11) (2015) 6298–6319.
[34] B. Barazandeh, K. Bastani, M. Rafieisakhaei, S. Kim, Z. Kong, M. A. Nussbaum, Robust sparse representation-based classification using online sensor data for monitoring manual material handling tasks, IEEE Trans. Automation Sci. Eng. (2017) 1–12.
[35] M. R. Mayiami, B. Seyfe, Nonparametric sparse representation, arXiv preprint arXiv:1201.2843.
[36] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Trans. Image Process. 15 (12) (2006) 3736–3745.
[37] L. Shao, R. Yan, X. Li, Y. Liu, From heuristic optimization to dictionary learning: A review and comprehensive comparison of image denoising algorithms, IEEE Trans. Cybernetics 44 (7) (2014) 1001–1013.
[38] K. Zhu, B. Vogel-Heuser, Sparse representation and its applications in micro-milling condition monitoring: noise separation and tool condition monitoring, Int. J. Adv. Manuf. Technol. 70 (1) (2014) 185–199.
[39] X.-S. Yang, Nature-inspired metaheuristic algorithms, Luniver Press, 2010.
[40] V. Stojanovic, N. Nedic, A nature inspired parameter tuning approach to cascade control for hydraulically driven parallel robot platform, J. Optim. Theory Appli. 168 (1) (2016) 332–347.
[41] V. Stojanovic, N. Nedic, D. Prsic, L. Dubonjic, V. Djordjevic, Application of cuckoo search algorithm to constrained control problem of a parallel robot platform, Int. J. Adv. Manuf. Technol. 87 (9-12) (2016) 2497–2507.
[42] D. Prsic, N. Nedic, V. Stojanovic, A nature inspired optimal control of pneumatic-driven parallel robot platform, Proc. Inst. Mechanical Eng., Part C: J. Mechanical Eng. Sci. 231 (1) (2017) 59–71.
[43] K. Ahmadi, E. Salari, Single-image super resolution using evolutionary sparse coding technique, IET Image Process. 11 (1) (2016) 13–21.
[44] P. M. Sheridan, F. Cai, C. Du, W. Ma, Z. Zhang, W. D. Lu, Sparse coding with memristor networks, Nature Nanotechnology 12 (2017) 784–789.
[45] M. Protter, M. Elad, Image sequence denoising via sparse and redundant representations, IEEE Trans. Image Process. 18 (1) (2009) 27–35.
[46] G. Peyre, Sparse modeling of textures, J. Math. Imag. Vis. 34 (1) (2009) 17–31.
[47] J. Mairal, M. Elad, G. Sapiro, Sparse representation for color image restoration, IEEE Trans. Image Process. 17 (1) (2008) 53–69.
[48] J. Mairal, G. Sapiro, M. Elad, Learning multiscale sparse representations for image and video restoration, SIAM Multiscale Model. Simul. 7 (1) (2008) 214–241.
[49] O. Bryt, M. Elad, Compression of facial images using the K-SVD algorithm, J. Visual Commun. Image Represent. 19 (4) (2008) 270–283.
[50] L. Peotta, L. Granai, P. Vandergheynst, Image compression using an edge adapted redundant dictionary and wavelets, Signal Process. 86 (3) (2006) 444–456.
[51] M. J. Fadili, J. L. Starck, F. Murtagh, Inpainting and zooming using sparse representations, Computer J. 52 (1) (2007) 64–79.
[52] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Discriminative learned dictionaries for local image analysis, in: Proc. IEEE Comp. Vis. Pattern Recog., 2008, pp. 1–8.
[53] H. Y. Liao, G. Sapiro, Sparse representations for limited data tomography, in: Proc. IEEE Int. Symp. Biomed. Imag., 2008, pp. 1375–1378.
[54] G. Davis, S. Mallat, M. Avellaneda, Adaptive greedy approximations, Constructive Approx. 13 (1) (1997) 57–98.
[55] A. M. Tillmann, On the computational intractability of exact and approximate dictionary learning, IEEE Signal Process. Lett. 22 (1) (2015) 45–49.
[56] S. S. Chen, D. L. Donoho, M. A. Saunders, Atomic decomposition by basis pursuit, SIAM J. Scientific Comput. 20 (1) (1998) 33–61.
[57] S. Mallat, Z. Zhang, Matching pursuit with time-frequency dictionaries, IEEE Trans. Signal Process. 41 (12) (1993) 3397–3415.
[58] Y. C. Pati, R. Rezaiifar, P. S. Krishnaprasad, Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition, in: Proc. Asimolar Conf. Sig. Sys. Compt., 1993, pp. 40–44.
[59] T. Blumensath, M. E. Davies, Gradient pursuits, IEEE Trans. Signal Process. 56 (6) (2008) 2370–2382.
[60] D. L. Donoho, For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution, Comm. Pure Applied Math. 59 (6) (2006) 797–829.
[61] Y. Nesterov, A. Nemirovskii, Interior-point polynomial algorithms in convex programming, SIAM, 1994.
[62] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Series B (1996) 267–288.
[63] B. Schwikowski, P. Uetz, S. Fields, A network of protein-protein interactions in yeast, Nature Biotech. 18 (12) (2000) 1257.
[64] C. A. Barnes, B. L. McNaughton, S. J. Mizumori, B. W. Leonard, L. H. Lin, Comparison of spatial and temporal characteristics of neuronal activity in sequential stages of hippocampal processing, Prog. Brain Res. 83 (1990) 287–300.
[65] E. Elhamifar, R. Vidal, Sparse subspace clustering, in: Proc. IEEE Comp. Vis. Pattern Recog., 2009, pp. 2790–2797.

[66] K. V. Mardia, P. E. Jupp, Directional statistics, Vol. 494, Wiley, 2009.
[67] K. V. Mardia, Statistics of directional data, Academic Press, 2014.
[68] R. Fisher, Dispersion on a sphere, in: Proc. R. Soc. Lond. A, Vol. 217, 1953, pp. 295–305.
[69] S. Zhong, Efficient online spherical k-means clustering, in: Proc. IEEE Neural Netw., Vol. 5, 2005, pp. 3180–3185.
[70] A. Banerjee, I. S. Dhillon, J. Ghosh, S. Sra, Clustering on the unit hypersphere using von mises-fisher distributions, J. Machine Learn. Res. 6 (2005) 1345–1382.
[71] S. Gopal, Y. Yang, Von mises-fisher clustering models, in: Proc. Int. Conf. Machine Learn., 2014, pp. 154–162.
[72] L. Meier, S. V. D. Geer, P. Buhlmann, The group lasso for logistic regression, J. R. Stat. Soc. Series B 70 (1) (2008) 53–71.
[73] J. Friedman, T. Hastie, R. Tibshirani, A note on the group lasso and a sparse group lasso, arXiv preprint arXiv:1001.0736.
[74] J. Huang, T. Zhang, The benefit of group sparsity, The Annals of Statistics 38 (4) (2010) 1978–2004.
[75] J. Huang, T. Zhang, D. Metaxas, Learning with structured sparsity, J. Mach. Learning Res. 12 (2011) 3371–3412.
[76] Y. C. Eldar, M. Mishali, Robust recovery of signals from a structured union of subspaces, IEEE Trans. Inf. Theory 55 (11) (2009) 5302–5316.
[77] R. Rubinstein, A. M. Bruckstein, M. Elad, Dictionaries for sparse representation modeling, Proc. IEEE 98 (6) (2010) 1045–1057.
[78] K. Engan, S. O. Aase, J. H. Husoy, Method of optimal directions for frame design, in: Proc. IEEE Acous. Speech Signal Process., Vol. 5, 1999, pp. 2443–2446.
[79] M. Aharon, M. Elad, A. Bruckstein, K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation, IEEE Trans. Signal Process. 54 (11) (2006) 4311–4322.
[80] T. Blumensath, M. Davies, Sparse and shift-invariant representations of music, IEEE Trans. Speech Audio Process." 14 (1) (2006) 50–57.
[81] P. Jost, P. Vandergheynst, S. Lesage, R. Gribonval, MoTIF: An efficient algorithm for learning translation invariant dictionaries, in: Proc. IEEE Acous. Speech Signal Process., Vol. 5, 2006, pp. 857–860.
[82] M. Aharon, M. Elad, Sparse and redundant modeling of image content using an image-signature-dictionary, SIAM J. Imaging Sci. 1 (3) (2008) 228–247.
[83] K. Engan, K. Skretting, J. H. Husoy, Family of iterative LS-based dictionary learning algorithms, ILS-DLA, for sparse signal representation, Digit. Signal Process. 17 (1) (2007) 32–49.
[84] P. Sallee, B. A. Olshausen, Learning sparse multiscale image representations, in: Adv. Neural Inf. Process. Syst., Vol. 15, 2003, pp. 1327–1334.
[85] S. Lesage, R. Gribonval, F. Bimbot, L. Benaroya, Learning unions of orthonormal bases with thresholded singular value decomposition, in: Proc. IEEE Acous. Speech Signal Process., Vol. 5, 2005, pp. 293–296.
[86] O. G. Sezer, O. Harmanci, O. G. Guleryuz, Sparse orthonormal transforms for image compression, in: Proc. IEEE Int. Conf. Image Process., 2008, pp. 149–152.
[87] R. Rubinstein, M. Zibulevsky, M. Elad, Double sparsity: Learning sparse dictionaries for sparse signal approximation, IEEE Trans. Signal Process. 58 (3) (2010) 1553–1564.
[88] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online learning for matrix factorization and sparse coding, J. Mach. Learning Res. 11 (1) (2010) 19–60.
[89] K. Skretting, K. Engan, Recursive least squares dictionary learning algorithm, IEEE Trans. Signal Process. 58 (4) (2010) 2121–2130.
[90] J. Mairal, F. Bach, J. Ponce, Task-driven dictionary learning, Tech. Rep. RR-7400, INRIA, France (2010).
[91] G. Monaci, P. Jost, P. Vandergheynst, Image compression with learnt tree-structured dictionaries, in: Proc. IEEE W. Mult. Signal Process., 2004, pp. 35–38.
[92] M. Nakashizuka, H. Nishiura, Y. Iiguni, Sparse image representations with shift-invariant tree-structured dictionaries, in: Proc. IEEE Int. Conf. Image Process., 2009, pp. 2145–2148.
[93] R. Jenatton, J. Mairal, G. Obozinski, F. Bach, Proximal methods for hierarchical sparse coding, J. Mach. Learning Res. 12 (2011) 2297–2334.
[94] J. Zepeda, Novel sparse representation methods; application to compression and indexation of images, Ph.D. thesis, INRIA, France (2010).
[95] J. Zepeda, C. Guillemot, E. Kijak, Image compression using sparse representations and the iteration-tuned and aligned dictionary, IEEE J. Selected Topics Signal Process. 5 (5) (2011) 1061–1073.
[96] I. Ramirez, P. Sprechmann, G. Sapiro, Classification and clustering via dictionary learning with structured incoherence and shared features, in: Proc. IEEE Comp. Vis. Pattern Recog., 2010, pp. 3501–3508.
[97] W. Dong, X. Li, L. Zhang, G. Shi, Sparsity-based image denoising via dictionary learning and structural clustering, in: Proc. IEEE Comp. Vis. Pattern Recog., 2011, pp. 457–464.
[98] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-d transform-domain collaborative filtering, IEEE Trans. Image Process. 16 (8) (2007) 2080–2095.
[99] D. A. Huang, L. W. Kang, Y. C. F. Wang, C. W. Lin, Self-learning based image decomposition with applications to single image denoising, IEEE Trans. Multimedia 16 (1) (2014) 83–93.
[100] R. Baraniuk, Compressive sensing, Lecture Notes IEEE Signal Process. Mag. 24 (4) (2007) 118–121.
[101] E. J. Candes, J. Romberg, T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, IEEE Trans. Inf. Theory 52 (2) (2006) 489–509.
[102] E. J. Candes, Y. C. Eldar, D. Needell, P. Randall, Compressed sensing with coherent and redundant dictionaries, App. Comput. Harmonic Anal. 31 (1) (2011) 59–73.
[103] R. G. Baraniuk, V. Cevher, M. F. Duarte, C. Hegde, Model-based compressive sensing, IEEE Trans. Inf. Theory 56 (4) (2010) 1982–2001.
[104] V. Cevher, P. Indyk, C. Hegde, R. G. Baraniuk, Recovery of clustered sparse signals from compressive measurements, Tech. rep., Rice Univ. (2009).
[105] L. Yu, H. Sun, J. P. Barbot, G. Zheng, Bayesian compressive sensing for cluster structured sparse signals, Elsevier Signal Process. 92 (1) (2012) 259–269.
[106] S. F. Cotter, B. D. Rao, K. Engan, K. Kreutz-Delgado, Sparse solutions to linear inverse problems with multiple measurement vectors, IEEE Trans. Signal Process. 53 (7) (2005) 2477–2488.
[107] J. Chen, X. Huo, Theoretical results on sparse representations of multiple-measurement vectors, IEEE Trans. Signal Process. 54 (12) (2006) 4634–4643.
[108] M. A. Davenport, M. F. Duarte, Y. C. Eldar, G. Kutyniok, Introduction to compressed sensing, Compressed Sensing: Theory and Applications, Cambridge Univ. Press, 2012.
[109] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, J. Royal Stat. Soc. B 68 (1) (2006) 49–67.
[110] Y. C. Eldar, P. Kuppinger, H. Bolcskei, Block-sparse signals: Uncertainty relations and efficient recovery, IEEE Trans. Signal Process. 58 (6) (2010) 3042–3054.
[111] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learning 20 (3) (1995) 273–297.
[112] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, Large margin methods for structured and interdependent output variables, J. Mach. Learning Res. 6 (Sep) (2005) 1453–1484.
[113] H. Cevikalp, B. Triggs, H. S. Yavuz, Y. Kucuk, M. Kucuk, A. Barkana, Large margin classifiers based on affine hulls, Neurocomput. 73 (16) (2010) 3160–3168.
[114] I. Steinwart, A. Christmann, Support Vector Machines, Springer, 2008.
[115] F. Girosi, An equivalence between sparse approximation and support vector machines, Neural Comput. 10 (6) (1998) 1455–1480.
[116] L. Xu, J. Neufeld, B. Larson, D. Schuurmans, Maximum margin clustering, in: Proc. Adv. Neural Inf. Process. Sys., 2004, pp. 1537–1544.
[117] K. Zhang, I. W. Tsang, J. T. Kwok, Maximum margin clustering made practical, IEEE Trans. Neural Netw. 20 (4) (2009) 583–596.
[118] J. R. Quinlan, Induction of decision trees, Mach. Learning 1 (1) (1986) 81–106.
[119] J. Haugeland, Artificial intelligence: The very idea, MIT Press, 1989.
[120] K. C. Gowda, E. Diday, Symbolic clustering using a new similarity measure, IEEE Trans. on Syst. Man Cybernetics 22 (2) (1992) 368–378.
[121] K. C. Gowda, E. Diday, Symbolic clustering using a new dissimilarity measure, Pattern Recog. 24 (6) (1991) 567–578.
[122] K. L. Du, Clustering: A neural network approach, Neural Netw. 23 (1) (2010) 89–107.
[123] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, Psychological review 65 (6) (1958) 386.
[124] F. Rosenblatt, Principles of neurodynamics. perceptrons and the theory of brain mechanisms, Tech. rep., CORNELL AERONAUTICAL LAB INC BUFFALO NY (1961).

[125] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Adv. Neural Info. Process. Sys., 2012, pp. 1097–1105.
[126] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Netw. 61 (2015) 85–117.
[127] J. R. Hershey, Z. Chen, J. L. Roux, S. Watanabe, Deep clustering: Discriminative embeddings for segmentation and separation, in: Proc. IEEE Acous. Speech Signal Process., 2016, pp. 31–35.
[128] P. Domingos, Master Algorithm, Penguin Books, 2016.
[129] Y. Oktar, M. Turkan, Dictionary learning with residual codes, in: Proc. IEEE Signal Proc. Comm. App. Conf., 2017, pp. 1–4.
[130] G. W. Milligan, M. C. Cooper, An examination of procedures for determining the number of clusters in a data set, Psychometrika 50 (2) (1985) 159–179.