

Sparse molecular image representation

Sofia Karygianni^a, Pascal Frossard^a

^a *Ecole Polytechnique Fédérale de Lausanne (EPFL), Signal Processing Laboratory (LTS4), CH-1015, Lausanne, Switzerland*

Abstract

Sparsity-based models have proven to be very effective in most image processing applications. The notion of sparsity has recently been extended to structured sparsity models where not only the number of components but also their support is important. This paper goes one step further and proposes a new model where signals are composed of a small number of molecules, which are each linear combinations of a few elementary functions in a dictionary. Our model takes into account the energy on the signal components in addition to their support. We study our prior in detail and propose a novel algorithm for sparse coding that permits the appearance of signal dependent versions of the molecules. Our experiments prove the benefits of the new image model in various restoration tasks and confirm the effectiveness of priors that extend sparsity in flexible ways especially in case of inverse problems with low quality data.

Keywords: sparsity, structure, linear combinations, deformations, pools, structural similarity

1. Introduction

Most tasks in signal processing and analysis are significantly simplified when the data is represented into its right form, especially for high-dimensional signals like images. The quest for the right signal representation has fostered the use of overcomplete dictionaries as tools for signal compression, denoising, enhancement and various other applications. Dictionaries have the advantage to have very few constraints in their construction, so that they can be

Email addresses: sofia.karygianni@epfl.ch (Sofia Karygianni),
pascal.frossard@epfl.ch (Pascal Frossard)

finally adapted to the data processing task at hand. However, this flexibility has a price: the representation of a signal is unfortunately not unique in overcomplete dictionaries, and finding the best such representation is generally an ill-posed problem. As a result, well-chosen priors or models about the signal representation become necessary in order to develop effective signal processing algorithms with overcomplete representations.

The most common models in overcomplete signal representations are based on sparsity priors. This means that the signal is well represented by only a few components or atoms of the overcomplete dictionary. Sparsity is a pretty intuitive prior that is also biologically plausible, as shown in the pioneer work of Olshausen and Field [1] where it is suggested that sparsity could be a property employed by the mammalian visual system for achieving efficient representations of natural images. Vast research efforts have been deployed in the last decades in order to design algorithms that solve the hard problem of sparse decomposition of signals by effective approximation [2, 3] or convex relaxation [4, 5].

While sparsity is a simple and generic model, it is not always a sufficient prior to obtain good signal reconstruction, especially if the original data measurements are compressed or inaccurate. More effective signal models can therefore be built by considering the dependencies between the dictionary elements that appear in the signal representation instead of their number only. In that spirit, group sparsity has been introduced as a way to enforce a pre-specified structure in the decomposition. Specifically, the components of the dictionary are partitioned into groups and the elements of each group are encouraged to appear simultaneously in the signal decomposition [6]. Alternatively, the atoms can also obey a predefined hierarchical structure [7]. Other approaches have considered additional flexibility by constraining the signal decomposition to include elements from overlapping groups of atoms [8, 9, 10]. The group sparsity structure is however not always appropriate for modeling signal patterns as the groups are merely identified in terms of their support. It is however not suitable for differentiating patterns with the same support but different distributions, which could actually be very different signal patterns. Such a case is presented in Figure 1 where we show how much the image of a face can change when varying the coefficients of its sparse code while keeping the same support. This ambiguity is unfortunately a serious drawback in various applications such as signal recovery and recognition, for example.

We propose here a new signal model to represent signal patterns and

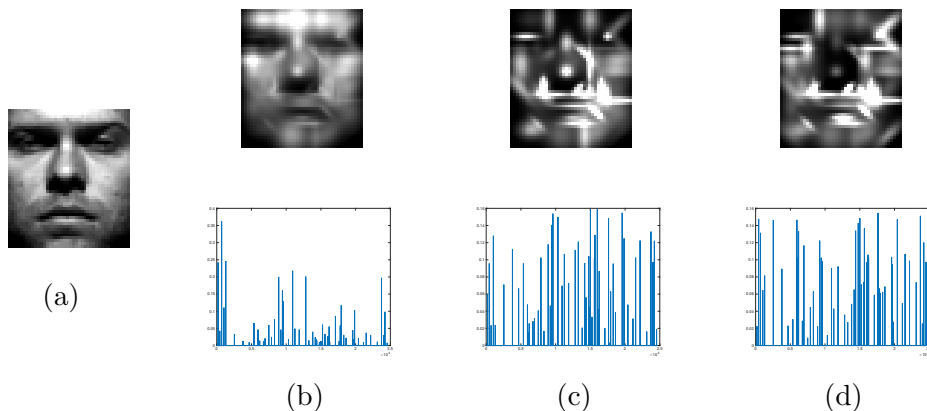


Figure 1: An example of the ambiguity related to the support of the sparse codes. In (a) we show the image of a face and in (b) its sparse approximation with 60 atoms on a dictionary of Gaussian atoms. The next two columns are produced by randomly choosing the values of the coefficients on the same support. The final signal is then normalized. The resulting images are quite different than the original face proving the importance of the coefficients along with the support of the sparse code.

higher level structures. Our model builds on our previous work on structured sparsity [11] and represents signals as sparse set of molecules, which are linear combinations of atoms from a redundant dictionary of elementary functions. It permits to efficiently represent the signal structures as parts or patterns; it builds richer priors than classical structured sparsity models that merely focus on the support of the signal representation and not the actual energy distribution. More specifically, we first form pools of similar atoms in the dictionary, and assume all atoms in a pool carry similar information. Then we define representative molecules that are prototype linear combinations of atoms, or equivalently typical patterns in images. Finally, we introduce the idea of molecule realizations in order to take into account the variability of patterns in natural images. The molecule realizations are slightly deformed versions of molecule prototypes, where atoms could be replaced by similar atoms from their respective pools. As a result, a given molecule can take various forms that are controlled by the construction of the atom pools. It provides flexibility in the representation of signals with molecules, while preserving the main structural information in the sparse signal approximation. The molecule prototype is essentially expressing a main visual pattern while its realizations allow for signal dependent versions of the main pattern with possibly minor deformations.

We study in details our new structured sparsity model and analyze the recovery performance of molecule representations. We formally show that our choice of the synthesis dictionary based on molecules realizations provides a good compromise between structure and flexibility. Then we propose a novel constructive sparse coding algorithm of signals with our new structured sparsity model. We exploit the characteristics of atoms pools to design effective similarity measures for detecting molecule realizations in signals. Finally, we show the use of our new framework with illustrative experiments in various applications such as compressed sensing, inpainting and denoising. Our results show that the new structured sparsity prior leads to better reconstruction performance than classical sparsity priors due to its flexible molecule-based representation.

Our efficient structured sparsity model represents a quite unique framework in the literature. In particular, the consideration of the coefficient distribution and the atom pools, as well as the definition of both molecule prototypes and realizations, are important characteristics of our new signal representation model. The coefficients permit to differentiate structures with distinct energy distributions on the same support and thus to facilitate the proper recovery of image information in case of incomplete or inaccurate observations. Another definition of molecule has been previously proposed in [12] to describe a set of coherent atoms in a dictionary, but it is more related to the notion of a group or a pool of atoms than to our original definition of a molecule. Multi-level structures are also related to the concept of double sparsity introduced in [13] where the authors learn structures on top of a set of predefined set of atoms. It is however less flexible than our model, where we include the notion of pools and molecules realizations that enable the proper handling of minor structure deformation in the signals. Less close to our model, some recent works describe the statistical dependencies between the atoms in a dictionary with graphical models. For example, Markov Random Fields (MRFs) are employed for modeling these dependencies in [14, 15, 16]. The resulting structure model is a probability distribution function that compares the different possible supports of atoms in the signal representation. These models are quite powerful but unfortunately quite complicated and highly parametric, such that they are difficult to deploy and adapt to various applications. Next, the idea of pooling that is used for defining molecules realizations is quite often used under different forms to provide local invariance [17, 18] in the signal representation. In our case however, it provides local invariance to small deformations of a set of

atoms with higher resilience to sparse code variability in the identification of typical patterns in images. Finally, the differentiation between the molecule prototypes and molecule realizations in our new model leads to realizations of structures that are signal dependent, like in [19, 20]. Hence, the signal representation is flexible but nevertheless follows a pre-defined structure. The specific characteristics of our scheme make it very suitable for various signal processing tasks and especially signal denoising and inpainting.

The structured sparsity model proposed in this paper is essentially a two-layer architecture with the first layer consisting of the dictionary atoms and the second of the molecules. The benefits of such architectures over the flat ones has been a subject of research for a long time in the feature extraction and machine learning community. It has been validated experimentally in the case of signal recognition in [21] while the mere existence of the field of deep learning can argue in benefit of multistage architectures. The deep learning systems consist of a hierarchy of features along with some pooling and contrast normalization operators that sequentially transform the input into a new representation [18, 22, 23, 20]. Although it is common to learn the filters used in each layer from the data, there is recent work done also in the case of predefined filters [24]. In both cases, the goal of the learning is to uncover class invariant signal representations that are mainly used for classification and not the learning of appropriate structure signal priors for signal recovery. These works nevertheless support the idea that multiple layers leads to better signal models, which is aligned with the ideas proposed in this paper.

In summary, we propose and study a novel two-layer signal model built on atoms and respectively molecules, where

- the inclusion of both the coefficients and the support in the structured sparsity prior enables the differentiation of structures on the same support but with distinct energy distributions
- the association of atoms with pools permits to be resilient to variability in the sparse codes
- the differentiation between the molecule prototypes and molecule realizations allows for additional flexibility for pattern recovery from incomplete or inaccurate observations

Besides the definition of a new structured sparsity model, this paper also provides the following contributions

- the design of new algorithm for sparse coding under our new structured sparsity prior
- the development of illustrative experiments in image recovery applications that confirm the benefits of our new framework.

The rest of the paper is organized as follows. In Section 2 we describe our model in detail. In Section 3 we compare the different choices for a suitable synthesis dictionary for our model while in Section 4 we present the associated coding problem in detail. Finally, in Section 5 we provide results that validate our model for various signal restoration tasks.

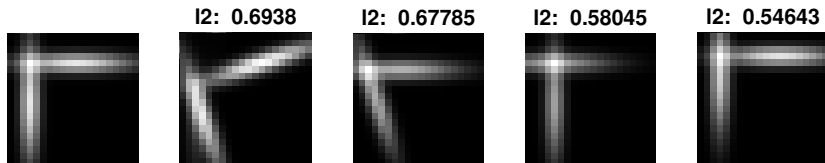
2. Structured image model

2.1. Multi-level structure

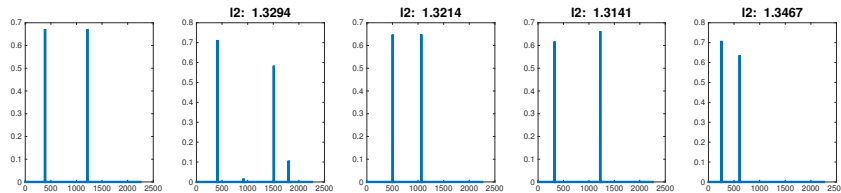
We present now our new structured sparsity model for images whose multi-level structure permits to represent visual patterns or typical signal parts as combinations of elementary atoms in a dictionary. In other words, we define molecules as linear combinations of atoms to represent groups of structurally similar signal patterns. We define the concept of molecule prototypes along with molecule realizations that are slightly deformed versions of the prototypes aiming at capturing additional signal variability. We first present our new model and then discuss in details the notion of pools of atoms, which is central for computing molecule realizations. Then we introduce a new structural difference function that is later used to compare visual patterns when computing image representations.

We first provide an example to illustrate our structured sparsity model. Our model is built on the concepts of molecule prototypes and realizations. The prototype is a representative pattern for a group of molecule realizations, which are slightly deformed versions of a typical image part. The first image in Figure 2a) shows a molecule prototype, which is an orthogonal angle formed by two edge-like atoms from the dictionary of elementary atoms. In other words, the molecule prototype is represented by a particular linear combinations of atoms, as shown in the first energy distribution function in Figure 2b. The molecule could however appear with small deformations in actual images, and such molecules realizations are illustrated in the rest of the images in Figure 2a. They look quite similar to the molecule prototype and preserve to some extent its structural characteristics, but they are not

constructed with the exact same atoms, as illustrated by their respective energy distribution functions in Figure 2b.



(a)



(b)

Figure 2: Illustrative example of a molecule prototype and its realizations. In (a) the molecule prototype (on the left) represents a near orthogonal crossing of edges while the molecule realizations describe visual patterns that are similar to the prototype. The l_2 distance between the prototype and the realizations in the image domain is given on top of each realization. In (b) we show the corresponding sparse codes of the images in (a). The l_2 distance of the sparse codes seen as vectors in \mathbb{R}^N is given on top of each figure. As we can see, none of the metrics depicts accurately the structural similarity among the patterns.

We now describe our new signal model in more details. We consider a set of signals $X \in \mathbb{R}^{N \times B}$ and a base dictionary $D \in \mathbb{R}^{N \times K}$ of elementary functions or atoms d_k with $1 \leq k \leq K$, whose linear combinations can effectively represent the signals X . We assume that the occurrence of atoms in the signal representation is not completely independent but that atoms rather have the tendency to form typical visual patterns. In other words, there are some linear combinations of atoms that tend to appear more frequently than others, possibly with slight changes either in the energy distribution or atom sets. The most frequent atom combinations are represented by a set of molecule prototypes $M = \{m_l, l \in \{1, \dots, Q\}\}$ where each prototype is defined

as a sparse set of atoms with specific coefficient values, i.e.,

$$m_l = \sum_{k=1}^K c_{\pi,l}(k) d_k = D c_{\pi,l}, \quad \|c_{\pi,l}\|_0 < n \quad (1)$$

where n is the sparsity level of the molecules and $c_{\pi,l}(k) > 0$ only if the atom d_k belongs to the support $\Gamma_{\pi,l}$ of the molecule m_l . The non-negativity of coefficients will be explained in more detail in Section 2.2. We can further write all the molecule prototypes in a matrix form as

$$M = D C_{\pi}, \quad \text{with } C_{\pi} = [c_{\pi,1} \ c_{\pi,2} \ \cdots \ c_{\pi,Q}]. \quad (2)$$

We consider that the molecules correspond to the most important parts in the signals, but that they may appear as realizations that are similar but not identical to the prototypes. Equivalently, we consider a signal $x \in X$ to be a sparse non-negative¹ combination of molecules realizations plus some bounded noise. We define $c_{x,l}$ as the vector of atom coefficients that expresses the realization of the molecule m_l in x . We further consider that the difference between a molecule realization and the corresponding prototype is small, i.e., $\Delta(c_{\pi,l}, c_{x,l}) < t, \forall l$, where the function Δ measures the structural difference between molecules. The parameter t is a threshold value on the structural difference and its value permits to control the flexibility of our new multi-level model in capturing the variability in typical visual patterns. The signal can therefore be written as

$$x = D C_x a + \eta, \quad \text{with } C_x = [c_{x,1} \ c_{x,2} \ \cdots \ c_{x,Q}] \quad \text{and} \quad \Delta(c_{\pi,l}, c_{x,l}) < t, \forall l \quad (3)$$

We further consider that the approximation error is bounded (i.e., $\|\eta\|_2 < H$), the atom and molecule coefficients are defined as $a_i \geq 0, \forall i$ and $c_{x,i}(k) \geq 0, \forall (k, i)$ and the representation is sparse, i.e., $\|a\|_0 \leq s$ for some sparsity threshold s .

The image model in Eq. (3) corresponds to a sparse decomposition of x into molecule realizations, or equivalently the expansion of the signal x into dictionary atoms whose coefficients are given by $C_x a$. The grouping of

¹In this level, we consider only positive coefficients to simplify the development, without loss of generality.

atoms into representative molecules is driven by the choice of the structural difference function Δ that quantifies the deviation of molecule realizations from the corresponding prototypes. In the rest of this section, we first introduce the concept of atom *pools*, which are groups of similar atoms in the dictionary, and eventually use atom pools to define the structural difference metric Δ that is used in our part-sparse signal model.

2.2. Pools of atoms

In our framework, the signal is represented as a linear combination of atoms taken from a redundant dictionary. The redundancy of the dictionary helps in building sparse representations but also leads to the fact that many atoms may carry similar information. In particular, a specific image feature can be well captured by a specific atom d_i in the dictionary. But the same feature might also be well represented by atoms that are similar to d_i , as illustrated in Figure 3. Depending on the actual image representation method, the same visual feature can therefore be coded in various ways. We would like to make sure that our part-sparse image model is able to take this phenomenon into account.

We define the notion of atom pools in order to represent atoms that are similar. More specifically, in a dictionary D , each atom d_i can be represented as a unit norm vector in the signal space \mathbb{R}^N . Then, there might be other atoms d_j in D that are very similar to d_i , i.e., $\langle d_i, d_j \rangle > 1 - \epsilon$, with ϵ the approximation threshold on the similarity of two atoms. In this case, the energy of the projection of d_j on d_i is significant, so that a visual feature may be equivalently well represented by the atoms d_i or d_j . We characterise this phenomenon by introducing the notion of pools of atoms: each atom d_i is related to a pool $P(d_i)$ of atoms d_j ’s that are most similar to d_i . In other words, a pool is defined as

$$P(d_i) = \{d_j, 1 \leq j \leq K, \mid \langle d_i, d_j \rangle > 1 - \epsilon\} \quad (4)$$

Equipped with this definition, we can now measure the difference between alternative representations of the same visual features. In particular, we can estimate the actual energy corresponding to the atom d_i in a signal represented by the sparse code b that does actually not include the atom d_i . In other words, looking at the sparse signal decomposition $x = Db$ with $b_i = 0$, we would like to know how much of the energy is actually aligned along the direction represented by the atom d_i . It mainly corresponds to the

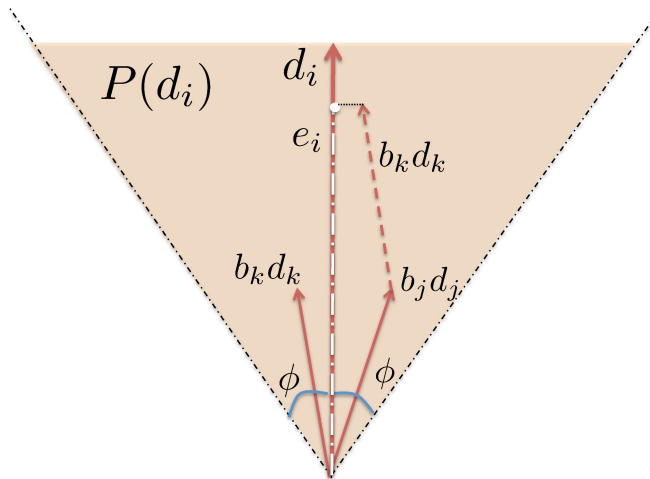


Figure 3: The representation of an atom d_i and its pool $P(d_i)$ in \mathbb{R}^N . The pool is defined by the atoms with $\cos \phi > 1 - \epsilon$. Then, $b_k d_k + b_j d_j$ is one possible realization of the atom d_i with energy $e_i = b_k \langle d_i, d_k \rangle + b_j \langle d_i, d_j \rangle$.

energy captured by the coefficient of all the atoms in the pool $P(d_i)$. We can therefore approximate the energy of the signal in the direction of d_i as

$$e_i(b) = \sum_{j \in P(d_i)} b_j \langle d_i, d_j \rangle = S_i b \quad (5)$$

where

$$S_i(j) = \begin{cases} \langle d_i, d_j \rangle & \text{if } d_j \in P(d_i) \\ 0 & \text{if } d_j \notin P(d_i) \end{cases} \quad (6)$$

The vector S_i expresses essentially the pairwise relationships between the atom d_i and the rest of the atoms in the dictionary D . The energy estimate above is very useful in computing the structural difference between molecules that is explained below. The value of $e_i(b)$ is essentially the length of the projection of the vector $v_i(b) = \sum_{j \in P(d_i)} b_j d_j$, the realization of d_i , in the direction of d_i . When the entries of b are non-negative, v_i is guaranteed to lie in the geometric space defined by the pool $P(d_i)$ and as a result the error $\|d_i - v_i\|_2^2$ is bounded (the proof is provided in Appendix A). In the rest, we will adopt this assumption of non-negativity without loss of generality. Finally, an example of the pool of an atom, as well as a possible non-negative realization of the atom from its pool, is shown in Figure 3.

2.3. Structural difference

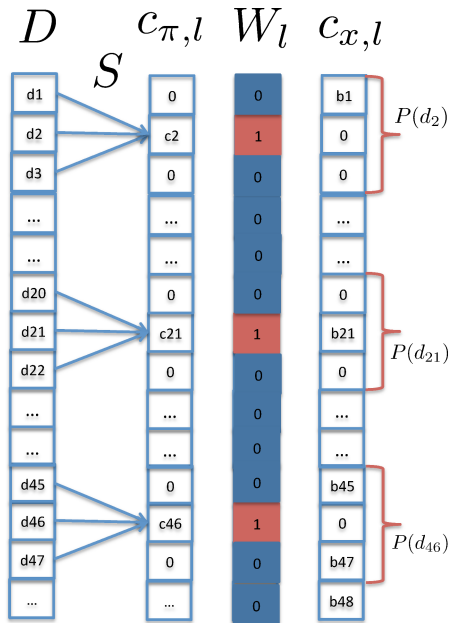


Figure 4: Illustration of a molecule prototype and a possible realization. The vector W_l is the indicator function of the support $\Gamma_{\pi,l}$ of the molecule prototype $c_{\pi,l}$. The structural difference between $c_{\pi,l}$ and $c_{x,l}$ is then $\Delta_{(c_{\pi,l}, c_{x,l})} = \|W_l \times (c_{\pi,l} - S c_{x,l})\|_2^2 = (c_2 - \langle d_1, d_2 \rangle b_2)^2 + (c_{21} - b_{21})^2 + (c_{46} - \langle d_{46}, d_{45} \rangle b_{45} - \langle d_{46}, d_{47} \rangle b_{47})^2$

We now propose a measure of structural difference between molecule instances that is based on the above definition of atom pools. Recall that a molecule realization is similar to a molecule prototype and permits to capture the variability of visual patterns in actual images. It can be defined as the deformation of a molecule prototype whose original atoms could be each substituted by atoms from their respective pool. Equivalently, a molecule realization is essentially a molecule prototype that can be realized through a linear combinations of atoms in the pools of the initial prototype components. As a result, a molecule realization has a similar energy as the prototype when measured on atom pools but not necessary exactly the same coefficient values on the atoms. It makes it difficult to measure the similarity between the actual visual patterns represented by the molecule prototype and its realizations. For example, the l_2 norm in both the image and sparse code domain fail to uncover the structural similarity between both molecules, as it does not take into account the actual features represented by the atoms nor their

interplay. The inability of the l_2 norm in capturing the similarity of molecules can be observed by checking the norms in Figures 2a and 2b.

As classical norm metrics are not appropriate for computing the similarity in the structure of different molecule instances, we propose a new structural difference $\Delta()$ for the signal model of Eq. (3). In particular, the deformation in the structure of molecules is measured by the compatibility between a sparse coefficient vector $c_{x,l}$ that represents the realization of the molecule m_l in the signal x , and the sparse coefficient vector $c_{\pi,l}$ that represents the corresponding molecule prototype. Since a molecule is identified by specific energy levels on the pools of the atoms in its support, its realizations are allowed to have non-zero values only in the union of the pools of these atoms, i.e., $\Gamma_{x,l} \subseteq \bigcup_{d_k \in \Gamma_{\pi,l}} P(d_k)$ where $\Gamma_{x,l}$ and $\Gamma_{\pi,l}$ are the supports of $c_{x,l}$ and $c_{\pi,l}$ respectively. Then, the structural difference computes the energy in the pools of $c_{x,l}$ and compares it to the ones expressed in $c_{\pi,l}$. If the energies are comparable, the structural difference is considered to be small.

To be more specific, using the formula for the energy level of an atom based on its pool given in (5), the structural difference Δ is computed as:

$$\begin{aligned} \Delta(c_{\pi,l}, c_{x,l}) &= \sum_{k|c_{\pi,l}(k)>0} (c_{\pi,l}(k) - e_k(c_{x,l}))^2 \\ &= \sum_{k|c_{\pi,l}(k)>0} (c_{\pi,l}(k) - S_k c_{x,l})^2 \\ &= \|W_l \times (c_{\pi,l} - S c_{x,l})\|_2^2 \end{aligned} \quad (7)$$

where $S = [S_1 S_2 \cdots S_K]$, with S_i from Eq. (6). The indicator vector W_l denotes the inclusion of dictionary atoms in the support $\Gamma_{\pi,l}$ of the molecule m_l , i.e.,

$$W_l(k) = \begin{cases} 1 & \text{if } d_k \in \Gamma_{\pi,l} \\ 0 & \text{if } d_k \notin \Gamma_{\pi,l} \end{cases} \quad (8)$$

Note that atoms that participate in the same molecule are assumed to not have overlapping pools which is equivalent to assuming that the atoms in a prototype are quite incoherent. As we will see in Section 3 this is a desired property that leads to lower coherence on the dictionary and thus better recovery guarantees. In general, the lower the structural difference $\Delta(c_{\pi,l}, c_{x,l})$, the more compatible the molecule realization and its prototype. Finally, we show an example of a molecule prototype and one possible realization in the

atomic level in Figure 4 along with the corresponding structural difference function.

3. Recovery Analysis

The proposed model presented in Eq. (1) defines signals to be formed as a composition of molecule prototypes with small, controlled deformations. The molecules are further defined as linear combinations of a set of basic atoms. According to this model, one could approximate signals in three different ways, namely as linear combinations of elements in three different dictionaries: the atomic dictionary D , the molecule prototype dictionary DC and the dictionary of molecule realizations. In the rest of this section, we analyze the pros and cons of each option in accurately representing signals.

On the one hand, the benefit of the atomic dictionary, is its flexibility since it includes all possible atoms present in signals. However, the lack of any structure makes it less appropriate for recovering signals under challenging conditions, in the presence of intense noise or when information is missing, as the sparsity prior may prove to be insufficient for a satisfactory reconstruction. On the other hand, it is known that the inclusion of more structure in the dictionaries facilitates significantly the task of signal restoration even under severe degradation. The dictionary of molecule prototypes as well as that of molecule realizations have both the advantage of providing structured priors. However, this advantage comes at a price in both cases.

The dictionary of molecule prototypes, might not be always sufficient for retrieving the right structure in the signals. We can rewrite a signal given from Eq. (1) as :

$$\begin{aligned} x &= D C_x a + \eta = D (C + E_x) a + \eta \approx D C a + D C \tilde{a} + \eta \\ &= D C (a + \tilde{a}) + \eta \\ &= D C b + \eta \end{aligned}$$

where $DC\tilde{a}$ is the best approximation of $DE_x\tilde{a}$ in the dictionary of molecule prototypes DC . The vector a is an exact sparse representation. However, E_x , which is the structured deviation from the prototypes, can take various forms so that the vector \tilde{a} does not necessarily have a sparse nature. Therefore, the structure of b can be significantly different from that of a resulting in a false recovery of the signal structure. The source of the above problem is the lack of flexibility in the dictionary DC : it defines patterns through the

prototypes to assist the retrieval of degraded signals but at the same time the dictionary elements are quite rigid and restrictive.

Therefore, it appears that building a dictionary with all possible molecule realizations, denoted as DC_x , could be a better and more flexible alternative with a compromise between structure and flexibility. However, building a dictionary with all possible molecule realizations results in a very coherent representation. As we have seen in Section 2.1, the molecule realizations are essentially small deformations of a molecule prototype. Therefore, all realizations of the same prototype are highly similar. The recovery performance of a dictionary is known to deteriorate as the sparsity of the signals decreases and the coherence of the dictionary increases. To put it more formally, a known recovery constraint for BPDN (Basis Pursuit Denoising) [25] or OMP (Orthogonal Matching Pursuit) [3] is given by

$$k \leq \frac{1}{2} \left(\frac{1}{\mu_x} + 1 \right). \quad (9)$$

where μ_x is the coherence of the underlying dictionary and k is the sparsity of the signal, i.e., the number of elements in the signal. Therefore, the more coherent the dictionary DC_x , the more sparse the signals should be in order to be able to recover them.

We can analyze how the coherence μ_x of the dictionary DC_x is affected by the presence of multiple realizations for each molecule prototype. Since the realizations of the same molecule prototype are very similar, μ_x can be lower bounded using the maximum distance r between any realization and the corresponding molecule prototype. The theoretical bound, $L_x \leq \mu_x$, is given by

$$L_x = 1 - 2r^2 \quad (10)$$

To quantify this result, we can compare the molecule realization dictionary with the case of a dictionary DC_u that contains only one molecule realization per molecule prototype. The restriction on the allowed number of instances per prototype allows for a theoretical upper bound on the coherence μ_u of the dictionary DC_u , i.e., $U_u \geq \mu_u$ with

$$U_u = \mu(1 - 2r^2) + 2r\sqrt{(1 - \mu^2)(1 - r^2)} \quad (11)$$

where μ is the coherence of the dictionary of molecule prototypes DC . In practice the coherence μ_u is expected to be close to μ . Both theoretical bounds depend on the distance r which is driven by the characteristics of the

atoms pools as well as the internal structure of the molecules. The latter is measured by the maximum similarity between atoms belonging to the same molecule, denoted as μ_M . To improve the readability of the section we have moved the exact expressions for r as well as the proofs for the bounds in the Appendix B.

From the expression for L_x we can see that the smaller the r is, the worse the μ_x is expected to be. On the other hand, when r is small, U_u gets closer to μ . In order to present these dependencies more concretely, we show in Figure 5 some plots of μ_x and μ_u for various settings. At the first row, we present the bounds L_x and U_u computed based on Eq. (10) and (11) respectively while at the second row we show the mean values of μ_x and μ_u computed experimentally for different values of the molecule prototype coherence μ over random generations of the dictionaries DC_u and DC_x . For simplicity, in our calculations we have assumed that the number of atoms in all molecules is the same, denoted as n . The pool angle ϕ was set to 10 degrees while we varied the maximum in-molecule atomic similarity μ_M . In both rows, the red line refers to the coherence of the DC_x dictionary, the blue line to the coherence of DC_u and the yellow to that of molecule prototypes DC .

From the figures, according to the values of the bounds L_x and U_u , the benefit of the use of DC_u over DC_x is more prominent when the molecule prototypes are not very coherent (lower values of μ). In this case, the lower bound for μ_x , L_x , is higher than the upper bound for μ_u , U_u , so that μ_u is guaranteed to be lower than μ_x . This benefit depends also on the coherence of the atoms belonging to the same molecules: it is larger when μ_M is low. However, the analysis of the experimental mean shows that in practice the coherence μ_u of the dictionary DC_u lies very close to the coherence of the initial molecule prototype dictionary DC , while μ_x lies always close to 1. Therefore, we observe that restricting the number of realizations in the dictionary to one per molecule prototype preserves the dictionary coherence quite well while the inclusion of more than one molecule realizations per prototype pushes the dictionary coherence towards 1.

To sum up, from the above discussion we can see that deciding which dictionary to use for signal decomposition is not trivial. The underlying atomic dictionary D lacks structure, the dictionary of molecule prototypes DC lacks flexibility while the dictionary of all molecule realizations suffers from inefficient size and high coherence. To alleviate this issue, we propose an iterative decomposition scheme that searches for the best molecule realizations using at each iteration a synthesis dictionary with strictly one molecule realization

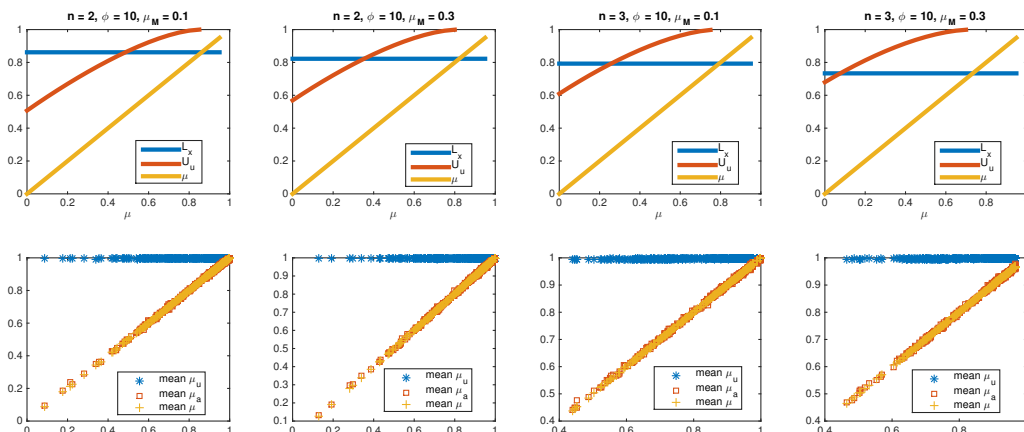


Figure 5: Comparison plots for the coherence of the dictionaries DC_x and DC_u containing many VS one realizations per molecule prototype respectively. The plots are for different values of the number of atoms per molecule n , the size of the atoms pools ϕ as well as the maximum similarity of atoms in the same molecule μ_M . In the first row we plot the theoretical bounds while in the second the average coherence observed over random generations of the dictionaries DC_x and DC_u .

per molecule prototype, denoted as DC_u above. In this way, at each iteration we have a guarantee for the coherence of the used dictionary while through the iterations we expect to recover the right signal structure. The details of the exact problem formulation as well as the proposed solution are presented in the next Section.

4. Adaptive molecule coding algorithm

We now formulate the problem of decomposing a signal into a sparse set of molecule realizations. We assume that the signal x follows the model in Eq. (3), or equivalently that the signal can be well approximated by a sparse linear combination of molecule realizations represented by C_x along with their respective coefficients a . Each molecule realization in C_x is a small deformation of the corresponding molecule prototype in C . The signal approximation can then be computed by solving the adaptive molecule coding problem written as follows:

$$\{\hat{a}, \hat{C}_x\} = \operatorname{argmin}_{a, C_x} \left[\|x - DC_x a\|_2^2 + \lambda_1 \|a\|_1 + \sum_{l, a(l) > 0} (\lambda_2 \Delta(c_{\pi, l}, c_{x, l}) + \lambda_3 \|c_{x, l}\|_1) \right] \quad (12)$$

where each $c_{x, l}$ is a molecule realization for the molecule prototype $c_{\pi, l}$ and $a(l) \geq 0$, $C_x(k, l) \geq 0 \forall l, k$. The first term in the objective function in Eq. (12) is the error of the approximation of the signal with a sparse set of molecule realizations. The second term favors a sparse approximation with the l_1 norm of the coefficient vector a . The last term drives the form of the molecule realizations: the term $\Delta(c_{\pi, l}, c_{x, l})$ tends to favor molecules realizations that are close to prototypes while the l_1 norm on the molecules realizations codes $c_{x, l}$ ensures their sparsity. The weight parameters λ_i 's permit to balance the different terms of the objective function.

By substituting the structural difference function from Eq. (7) in Eq. (12) we get:

$$\{\hat{a}, \hat{C}_x\} = \operatorname{argmin}_{a, C_x} \left[\|x - DC_x a\|_2^2 + \lambda_1 \|a\|_1 + \lambda_2 \sum_{l, a(l) > 0} (\|W_l \times (c_{\pi, l} - S c_{x, l})\|_2^2 + \lambda_3 \sum_{l, a(l) > 0} \|c_{x, l}\|_1) \right] \quad (13)$$

where W_i is given in Eq. (8) and the set of λ_i 's are weight parameters. For a given dictionary D , a set of pools represented by S and a set of molecule prototype written as C_π , the objective function F_{D, S, C_π} is minimized when the variables a and C_x form a part-sparse approximation of x . However, the above optimization problem cannot unfortunately be solved easily for both variables a_x and C_x at the same time. Clearly, the problem is not jointly convex for both variables. However, when one of the variables is fixed, the problem is convex with respect to the other one. Therefore, we adopt an alternating optimization technique with two steps for solving the optimization problem in Eq. (13). The two steps are computed as follows.

1. We first fix the set of molecules realizations, and solve the sparse coding problem for the coefficient vector a . Given C_x , the solution for a can be found as:

$$\hat{a} = \operatorname{argmin}_a [\|x - DC_x a\|_2^2 + \lambda_1 \|a\|_1], \text{ with } a(i) \geq 0, \forall i \quad (14)$$

2. Then, we fix the coefficient vector, and find the set of molecule realizations that minimize the objective function of the coding problem. Given a , the solution for C_x can be found as

$$\hat{C}_x = \operatorname{argmin}_{C_x} \left[\|x - DC_x a\|_2^2 + \lambda_2 \sum_{l, a(l) > 0} (\|W_l \times (c_{\pi, l} - S c_{x, l})\|_2^2 + \lambda_3 \sum_{l, a(l) > 0} \|c_{x, l}\|_1) \right], \quad (15)$$

with $c_{x, l}(k) \geq 0, \forall l, k$

The first problem is essentially an l_1 regularized sparse coding problem which is convex with a . It can be solved with many different algorithms, e.g., [2, 26]. In our case we have chosen to solve it with the method of alternating direction method of multipliers (ADMM) [27]. Following the findings in [28], we also employ the method of reweighted l_1 -minimization as it leads to a sparser solution. Note that, at the very first iteration of the global algorithm, C_x is initialized with C_π , while it is later updated during the solution of the second step of the alternating algorithm.

The second problem is also convex. As for the first problem, we have chosen to solve it with ADMM [27]. In order to solve it more efficiently, we however rewrite it so that it is optimized for one vector of coefficients b instead of the matrix C_x . As explained in Section 2.3, the support of each molecule realization is restricted to the union of the pools of the active atoms in the corresponding molecule prototype. Therefore, many of the entries in matrix C_x are constrained to be zero. The vector b represents the possible non-zero entries in C_x , i.e. the coefficients of the atoms that are part of the pools of the active atoms in the molecules that compose x (given in a). Essentially it expresses the flexibility that is allowed in the molecule realizations once the molecules are chosen.

To complete our problem transformation, we further introduce the vector \tilde{C} that expresses the expected energy in the atoms pools. It is created

by concatenating into a vector form the entries in C that correspond to the energy expected in each pool of active atoms. Then, we also create a corresponding dictionary of atoms \tilde{D} by concatenating the atoms in each of the active pools. Finally, the new vector of relationships \tilde{S} between atoms in \tilde{D} replaces the vector S . With these modifications, the problem in Eq. (15) can be equivalently expressed as:

$$\hat{b} = \underset{b}{\operatorname{argmin}} \|x - \tilde{D}b\|_2^2 + \lambda_2 \|\tilde{C} - \tilde{S}b\|_2^2 + \lambda_3 \|b\|_1 \text{ with } b(k) \geq 0, \forall k \quad (16)$$

The solution of this problem is much less time consuming than the one of the equivalent problem in Eq. (15) as the size of the vector b is usually much smaller than that of the whole dictionary D .

Finally, we iterate between the two optimization problems until the value of the signal reconstruction doesn't change much. Although this alternate optimization technique does not have any optimality guarantee, it gives good results in practice and therefore offers an effective constructive solution to the sparse coding problem of Eq.(13). Since the algorithm has several constraints on the structure and sparsity the final molecule realizations cannot be completely different from the predefined molecule prototypes and as a result the quality of the signal reconstruction depends significantly on the initialization of the molecule structure. However, the design and learning of good molecule prototypes is beyond the scope of this paper which is mainly focused on the sparse coding step and remains as interesting future work. Finally, as long as the parameters of the algorithm are concerned, the values for the λ 's were chosen according to each specific task based on a small validation set. The value for the parameter r required for the ADMM method was set to 1 for all the experiments. The pseudocode of the complete sparse coding scheme, called Adaptive Molecule Coding (AMC), is presented in Algorithm 1.

5. Experimental results on signal restoration

Next, we have evaluated the effectiveness of our model for various image restoration tasks on both synthetic data and real data. In signal restoration, a high quality signal x needs to be reconstructed from its degraded measurements y . The problem can be modeled in a generic form as

$$y = Hx + v \quad (17)$$

where H is a degrading operator and v is additive noise.

Algorithm 1 Adaptive molecule coding (AMC)

Input: $x, D, C_\pi, S, \lambda_1, \lambda_2, \lambda_3, \epsilon$

- 1: $\hat{a} = \operatorname{argmin}_a [\|x - D C_\pi a\|_2 + \lambda_1 \|a\|_1], a \geq 0$ \triangleright Initialize a
- 2: **while** true **do** \triangleright Alternate optimization
- 3: $(\tilde{D}, \tilde{S}, \tilde{C}) = \operatorname{transform}(D, C, S, \hat{a})$ \triangleright Create new variables for Eq. (16)
- 4: $\hat{b} = \operatorname{argmin}_b [\|x - \tilde{D} b\|_2^2 + \lambda_2 \|\tilde{C} - \tilde{S} b\|_2^2 + \lambda_3 \|b\|_1], b \geq 0$ \triangleright Solve for b
- 5: $\hat{C}_x = \operatorname{transform}^{-1}(\hat{b}, C, \hat{a})$ \triangleright Reconstruct C_x from b
- 6: $w = 1./\hat{a}$ \triangleright Set new weights for re-weighted l_1
- 7: $\hat{a} = \operatorname{argmin}_a [\|x - D \hat{C}_x a\|_2 + \lambda_1 \|w * a\|_1], a \geq 0$ \triangleright Solve for a
- 8: **if** $\hat{C}_x \hat{a} - C_p a_p < \epsilon$ **then** return \triangleright If signal coding did not change significantly, stop
- 9: **else**
- 10: $a_p = \hat{a}, C_p = \hat{C}_x$
- 11: **end if**
- 12: **end while**

Output: \hat{a}, \hat{C}_x

5.1. Synthetic Data

To be more specific, we have used a dictionary of gaussian anisotropic atoms to create signals of size 10×10 . The mother function of the dictionary can be written as $\phi(x, y) = A \exp(-(x/2)^2 - y^2)$ and we have sampled the image plane for two scale levels $[0.5 \ 1]$ with a step size 1 for translation and $\pi/6$ for rotation. The atoms of the dictionary were combined according to 10 predefined molecules contained in C . Each molecule was randomly constructed to contain 2, 3 or 4 atoms of equal energy. Then each signal was created as a random combination of a few molecule realizations (2, 3 or 4). To produce a molecule realization we used the following procedure: for each atom in the molecule prototype we produced an approximation using the atoms in the atom’s pool. The atoms were chosen randomly, their total number drawn from a geometric distribution with $p = 0.7$ (so that the approximation is a sparse combination of atoms) while their coefficients were adjusted so that the projection of their combination to the atom direction is close to the original coefficient value. Finally, for each restoration task, the appropriate operator was applied to get the testing data.

We compare our method with the l_1 - l_2 group norm (the algorithm is denoted as A12 in this paper) [10]. Each group $g_i \in \mathcal{G}$ is defined according to the support of the corresponding molecule m_i . If an atom doesn’t belong to any group, then it is considered a group by itself. The equivalent optimization problem is:

$$\hat{b} = \underset{b}{\operatorname{argmin}} \{ \|y - H D b\|_2 + \lambda \sum_{g_i \in \mathcal{G}} \|b_{g_i}\| \} \quad (18)$$

The vector of coefficients b is the signal decomposition in the atomic level and b_{g_i} is its restriction on g_i . The decomposition \hat{a} in groups (or equivalently molecules in our case) is computed as the l_2 norm of the coefficients in each group i.e., $\hat{a}_i = \|\hat{b}_{g_i}\|$. As we have discussed before, one alternative for the synthesis dictionary is the dictionary of molecules prototypes. This approach is similar to the sparse coding step in [13]. In the following, we also compare our scheme with sparse coding with l_1 regularization on the molecule dictionary (we denote this algorithm as Am), i.e., the outcome of:

$$\hat{a} = \underset{a}{\operatorname{argmin}} \{ \|y - H * D_m * a\|_2 + \lambda \|a\|_1 \} \quad (19)$$

where $D_m = DC$ is the molecule dictionary. Finally, we also compare against

simple sparse coding on D , i.e.,

$$\hat{a} = \underset{a}{\operatorname{argmin}} \{ \|y - D * a\|_2 + \lambda \|a\|_1 \} \quad (20)$$

This approach is denoted as A1 in the rest of the paper.

The performance of the algorithms is compared using various measures. To quantify the performance in terms of the signal recovery we compute both the mean square error of the signal approximation (MSE), i.e., $\frac{\sum_i \|x_i - \hat{x}_i\|^2}{N}$ where \hat{x} is the signal reconstruction and N is the number of signals, as well as the mean sparsity ratio of the recovered representations where the sparsity ratio is computed as the l_0 norm of the recovered representation over D over the l_0 norm of the true atomic representation. Moreover, we are also interested in how effective are the schemes in detecting the correct molecules. Therefore, we also provide results for the accuracy of the molecule detection, which is the ration of the correctly categorized molecules ($TP + TN$) over all the molecule instances ($P + N$).

5.1.1. Denoising

Firstly, we have tested the performance of the schemes under noise. In this case, $H = I$ and v is white gaussian noise. The results, for different noise levels, are shown in Figure 6. For each noise level, the results were averaged over 5 different molecule matrices and 1000 signal instances per matrix. The parameters for each algorithm, chosen based on a small validation set, were: $l_1 = 0.01, l_2 = 1, l_3 = 0.1$ for AMC and $l_1 = 0.1$ for all the rest. From Figure 6 we can observe that as the noise increases the effectiveness of the structure is more prominent: the MSE of A1 progressively deteriorates compared to the other 3 schemes that use a structured prior. Moreover, for the highest noise level the Am scheme which is the one with the least flexible structure prior, almost reaches the best performance. However, our scheme manages to perform best for all the noise levels by uncovering signal representations with small MSEs, accurate molecule detection, and satisfactory sparsity (Am has a fixed sparsity level for each molecule, therefore it is expected to have the lower value as the most constrained one).

5.1.2. Inpainting

Next, we have tested the performance of the schemes for inpainting. In this case, we have created a set of signals by omitting the signal values in a randomly chosen square region. We have tried three different sizes for the

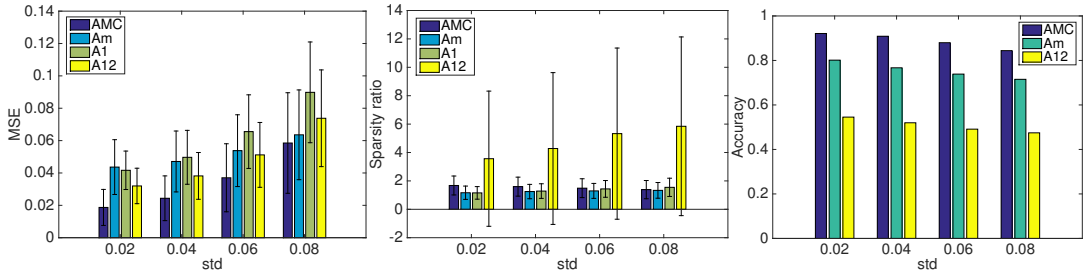


Figure 6: Results for denoising

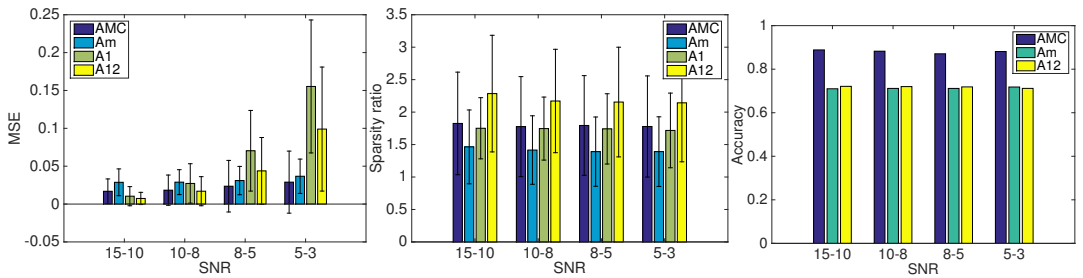


Figure 7: Results for Inpainting

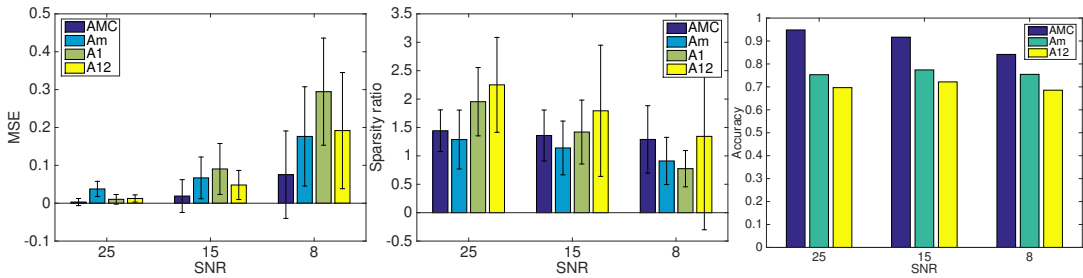


Figure 8: Results for CS

region: 3×3 , 4×4 and 5×5 . Then, the signals were divided into 4 sets based on their SNR. The signal recovery problem was solved over the known regions of the signals: each signal x was expressed as $x' = P_x * x$ where P_x is the mask denoting the known region. In this case, $H = P_x * I$ resulting in masking each dictionary atom. No extra noise was added to the data. The values for the parameters were $l_1 = 0.001, l_2 = 1, l_3 = 0.1$ for AMC and $l_1 = 0.01$ for all the rest. The results are shown in Figure 7. Again, we can observe the benefits from the flexible prior that our scheme provides compared to the rest: the MSE is always the lowest, the accuracy is the

highest while the sparsity ratio is satisfactory, usually the lowest after Am which is the most constrained one. In case of highly disturbed signals (lowest SNR) the Am also outperforms the rest, proving the importance of structure in applications where there is a significant amount of missing information.

5.1.3. Compressed Sensing

Finally, we have compared the recovery performance of the schemes for compressed sensing. The measurement process was performed by setting $H = \Phi$ where Φ is a random projection matrix. The entries of Φ were independent realizations from a standard normal distribution. We have checked three different sizes for Φ namely 25, 15 and 8 measurements. For each number of measurements the results were averaged over 5 different instances of matrix Φ . The values of the parameters were $l_1 = 0.01, l_2 = 10, l_3 = 0.01$ for AMC and A1 while $l_1 = 1$ for Am and $l_1 = 0.01$ for A12. The results for the different number of measurement are shown in Figure 8. Our scheme significantly outperforms the rest as the number of measurements decreases while keeping a high accuracy on molecule detection. The sparsity ratio is almost stable for all sizes of measurement matrix and quite close to 1 which is the desired value.

5.2. Denoising of digit images

Next, we have used our adaptive molecule coding scheme to perform denoising on MNIST images [29]. The images have been downsampled to 14×14 and normalized. In order to better fit the signal model the digits were further coarsely pre-aligned to avoid big discrepancies in the position and the orientation. The molecule prototypes were extracted using the algorithm presented in [13] from 1000 examples per digit while for the testing we used 100 examples per digit. The denoising performance was tested over different noise levels and measured by the mean squared error and the mean sparsity ratio. The parameters were fixed according to a small validation set and their values were $l_1 = 0.001, l_2 = 0.01, l_3 = 0.01$ for AMC and $l_1 = 0.01$ for the rest of the schemes.

The results of our experiments are presented in Figure 9. We have experimented with both denoising each digit separately using molecules extracted only for its class as well as denoising with molecules extracted from many classes simultaneously. In the first two columns we show the results we obtained for digits 0 and 9 separately while in the third column we plot the results for the case of denoising digits 0, 1, 2 and 3 with molecules extracted

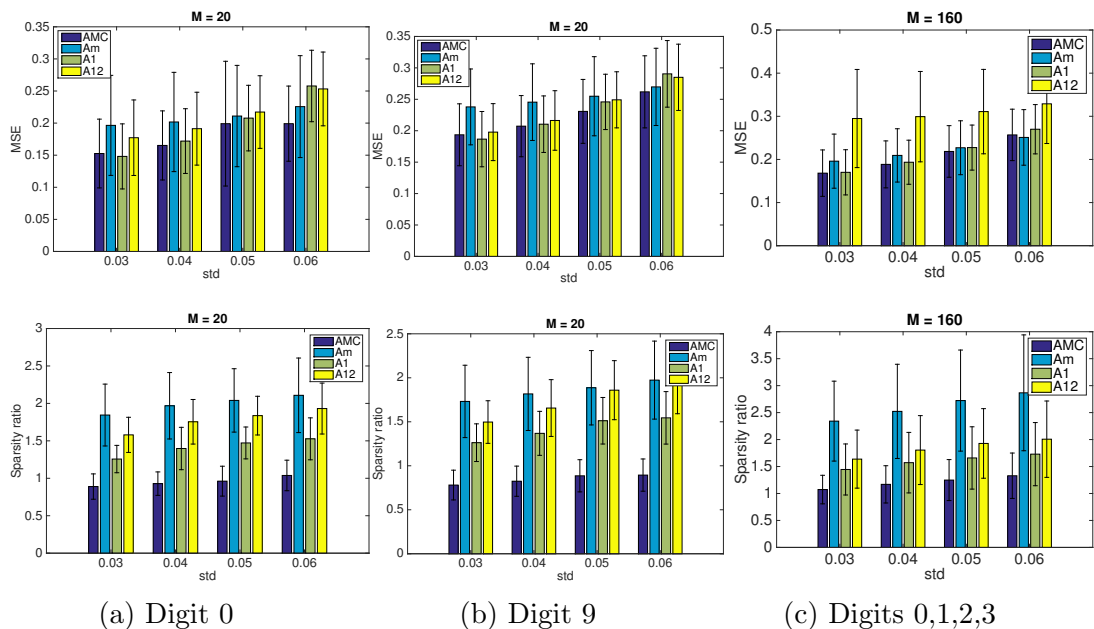


Figure 9: Results for denoising on data from MNIST digits for various levels of noise. On the first row we plot the MSE and on the second the sparsity ratio of the results. In the first two columns we present the results obtained when each digit was treated separately while on the third row we simultaneously denoised digits from different classes. The number of used molecules M is written in the title of each figure.

for all 4 digits together. From the plots we can see that AMC is the scheme that manages to perform well for all different noise levels. As expected the benefits from rich structure priors are more prominent in the presence of severe noise, where Am, the scheme with the most restrictive prior, outperforms A1 and A12 that have looser priors. However, for lower noise levels the performance of Am is not sufficiently good due to the rigidity of its prior. Our scheme on the other hand performs well in all cases as it adapts to the signals almost as successfully as A1 in the less noisy cases, while preserving the structure as Am in the more noisy cases. Finally, it is also important to note that AMC is the scheme that achieves on average a sparsity ratio close to one, meaning that it is highly efficient as it achieves a good signal restoration using only as many components as it is necessary.

5.3. Restoration of image patches

Finally, in image restoration it is often the case that the non-local similarities that different regions of the image may exhibit are used to enhance

the restoration process [30, 31]. The idea of ‘nonlocally centralized’ sparse codes is not very far from the idea of molecule prototypes. Therefore, we have followed the same intuition to define molecules prototypes based on the non-local similarity of patches and use their deformed versions to further enhance the image recovery from compressed measurements

To be more specific, when only sparsity is used as a prior for the recovery of the patches x_i of an image X the recovery problem for each patch can be written as:

$$\hat{a} = \operatorname{argmin} \|y_i - \Phi D a_i\|_2^2 + \lambda_1 \|a_i\|_1 \quad (21)$$

where a_i is the decomposition of the patch x_i in the dictionary D and y_i is the measurements acquired for this patch. The recovered image created by the recovered patches \tilde{x}_i is then \tilde{X} .

However, taking into account the non-local similarity of the patches one can extract a molecule prototype for every patch and further enhance the recovery by restricting the code of the each patch to be a realization of the prototype. The corresponding coding problem is then:

$$c_{\hat{x}_i} = \operatorname{argmin} \|y_i - \Phi D c_{x,i}\|_2^2 + \lambda_2 \|W_i \times (c_{\pi,i} - S * c_{x,i})\|_2^2 + \lambda_3 \|c_{x,i}\|_1 \quad (22)$$

where $c_{\pi,i}$ is the molecule prototype for \tilde{x}_i and $c_{x,i}$ is the patch dependent molecule realization. In order to obtain $c_{\pi,i}$ we search the image \tilde{X} for the most similar patches to \tilde{x}_i and we build a set Ω_i as in [30]. Then, based on the sparse codes of the patches in Ω_i we extract a molecule prototype for \tilde{x}_i . The prototype extraction algorithm is a greedy procedure that identifies a small number of atoms to account for most of the energy in the sparse codes in Ω_i while taking into account the atoms pools. It is an iterative procedure that at each step adds in the support of the molecule prototype the atom with the most energy in its pool. The energy of the atoms falling in the already chosen pools is considered covered and the algorithm iterates until a sufficient amount of the energy is covered. In this way, we extract a molecule prototype $c_{\pi,i}$ that accepts as realizations all the patches in Ω_i .

To show that our proposed coding scheme is suitable for enhancing the recovery of the original image, we have compared it to the l_1 based sparse coding presented in Eq. (21) which only imposes sparsity as structure. Moreover, following the ideas in [30], we have also implemented a scheme where the imposed structure is defined as the mean sparse code over similar patches. The corresponding optimization problem is then:

$$\hat{a} = \operatorname{argmin} \|y_i - \Phi D a_i\|_2^2 + \lambda_2 \|\hat{a}_i - a_i\|_2^2 + \lambda_3 \|a_i\|_1 \quad (23)$$

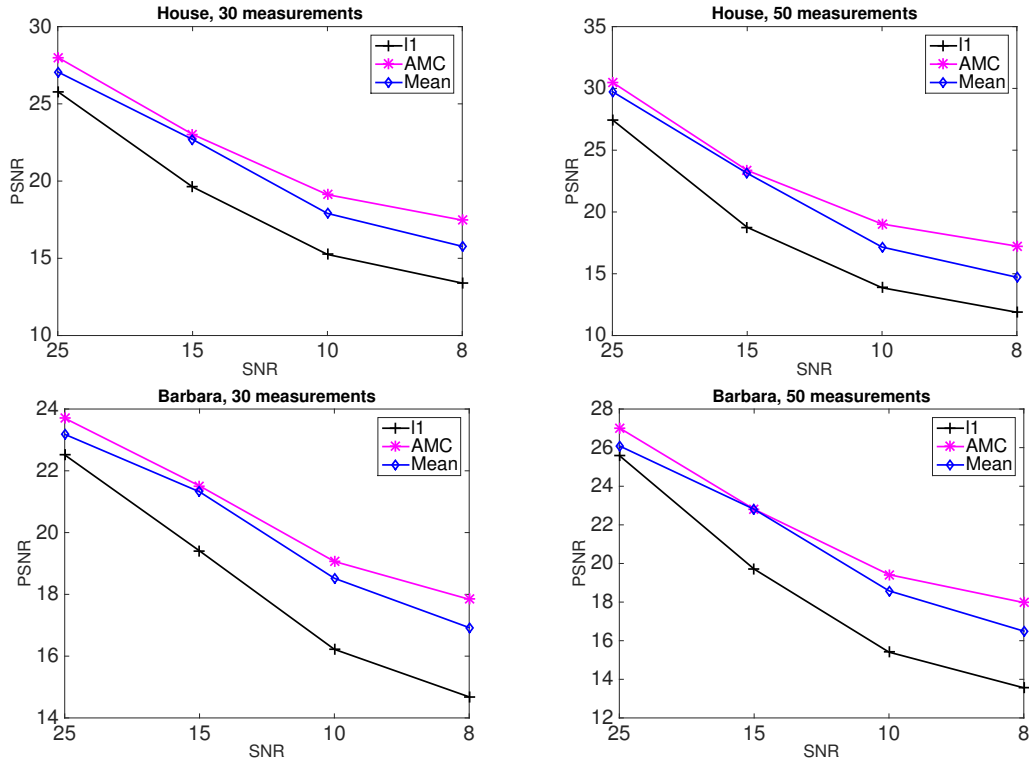


Figure 10: Results for image recovery with compressed measurements. The values of the parameters were set to $l_1 = 10$ and $l_2 = l_3 = 1000$.

where \hat{a}_i is the mean sparse code obtained from the sparse codes of the patches in Ω_i .

We have tested the performance of the above schemes on the images ‘House’ and ‘Barbara’. Each image was divided in 10×10 , non-overlapping patches. As a base dictionary D we have used a DCT overcomplete dictionary with 256 atoms. For solving the coding problem in Eq. (22) we have used the Algorithm 1, namely the part that solves for C_x given a , as in this case for each patch there is only one molecule prototype and as result the vector of molecule coefficients is set to 1. The entries of Φ were independent realizations from a standard normal distribution. We have checked two different size for Φ , namely 30 and 50 measurements, while for each number of measurements the results were averaged over 5 different instances of the matrix Φ . The measurements were further corrupted with noise.

In the Figure 10 we show the PSNR of the recovered images based on the three different schemes for various levels of noise and the two different number of measurements. From the results we can verify that the non-local similarity of the patches is very helpful for the image restoration as the l_1 sparse coding has a much lower PSNR than the other two schemes. Moreover, our molecule based coding scheme manages to extract more effectively the structural similarities of the patches than the mean sparse code as it achieves better PSNR results for the majority of settings. Therefore it is proven that the idea of molecule prototypes and realizations based on atoms pools is a powerful one providing correct priors for patch based restoration of images.

6. Conclusions

In this paper we have presented a new two-layer structure model for signals. We have defined our structural elements, the molecules, as linear combinations of atoms and we have distinguished between molecule prototypes and molecule realizations based on the notion of pools of atoms. The addition of coefficients in the structure permits a better modeling of higher level patterns while the definition of molecule realizations results in extra invariance to small deformations of patterns. We have presented our new algorithmic scheme for adaptive molecule coding (AMC) and we have conducted experiments on both synthetic and real data that proved the effectiveness of our model for various restoration tasks.

7. Acknowledgments

We would like to thank Prof. Christine Guillemot for the fruitful discussions and comments. This work was been partially fundend by the Swiss National Science Foundation (SNSF) under Grant 200021_143999.

Appendix A. Bound on error of atom realization

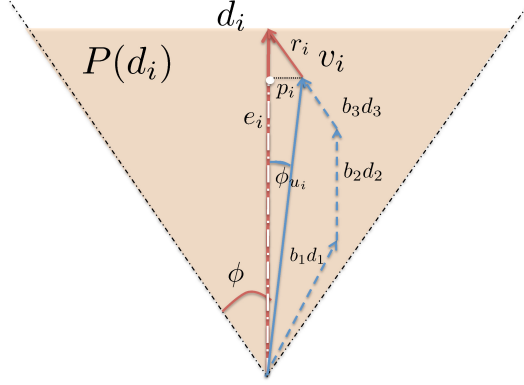


Figure A.11: An example of the realization of the atom d_i from vector $v_i = b_1 d_1 + b_2 d_2 + b_3 d_3$ with $d_1, d_2, d_3 \in P(d_i)$ and $b_1, b_2, b_3 > 0$.

As we have mentioned in Section 2.2, if we constrain the atoms that participate in the realization of the atom d_i to lie in its pool $P(d_i)$ and have non-negative coefficients we can guarantee that the resulting approximation has a bounded error, i.e., $\|d_i - v_i\|_2^2 \leq L$. To see why, let $v_i = \sum_{j \in P(d_i)} b_j d_j$. Then, from Figure A.11 we have:

$$\|d_i - v_i\|_2^2 = \|r_i\|^2 = \|p_i\|^2 + (1 - e_i)^2 = e_i^2 \tan^2 \phi_{u_i} + (1 - e_i)^2 \quad (\text{A.1})$$

However for the angle between v_i and d_i we have:

$$\cos \phi_{u_i} = \frac{\langle v_i, d_i \rangle}{\|v_i\|} = \frac{\sum_{j \in P(d_i)} b_j \langle d_j, d_i \rangle}{\|\sum_{j \in P(d_i)} b_j d_j\|} \geq \frac{(1 - \epsilon) \sum_{j \in P(d_i)} b_j}{\sum_{j \in P(d_i)} |b_j|} = 1 - \epsilon$$

if $b_j \geq 0, \forall j \in P(d_i)$. Therefore, when we allow only non-negative coefficients in the approximation, v_i belongs in $P(d_i)$.

Moreover, since $\cos \phi_{u_i} \geq 1 - \epsilon$, then $\sin \phi_{u_i} \leq \sqrt{\epsilon(1 - \epsilon)}$ and therefore $\tan \phi_{u_i} \leq \sqrt{\frac{\epsilon}{1 - \epsilon}}$. Finally, from Eq. A.1 we get:

$$\|d_i - v_i\|_2^2 \leq (1 - e_i)^2 + e_i^2 \frac{\epsilon}{(1 - \epsilon)} \quad (\text{A.2})$$

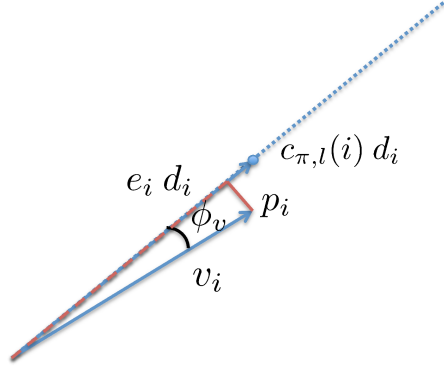


Figure B.12: An example of the approximation of the atom d_i from vector v_i deviating by ϕ_v in direction. The desired energy level is c_{li} while the projection of v_i gives an energy of e_i .

Appendix B. Recovery analysis supplementary material

We now present the theorems that provide the lower and upper bounds on the coherence of dictionaries DC_x and DC_u discussed in Section 3. The dictionary DC_x is a dictionary that contains more than one realizations per molecule prototype while the dictionary DC_u is restricted to one realization per prototype. To evaluate their coherences denoted as μ_x and μ_u respectively we will first need to examine the distance between a molecule prototype $m_{\pi,l} = D c_{\pi,l}$ and its possible realizations $m_{x,l} = D c_{xi,l}$. The corresponding upper bound is presented in the next Theorem.

Theorem 1

Let $\|m_{\pi,l}\|_0 \leq n, \forall l$ and $\phi = \text{acos}(1 - \epsilon)$ where ϵ is the parameter used in the pool definition in Eq. (4). Moreover, let the error $|c_{\pi,l}(i) - e_i|$ between the energy in an atom d_i of a molecule prototype and the energy on its pool on any of the molecule realizations be bounded by $|c_{\pi,l}(i) - e_i| \leq E c_{\pi,l}(i), \forall l, i \in \Gamma_{\pi,l}$ where E is a positive constant. Finally, let μ_M to stand for the in-molecule coherence defined as the maximum coherence between the atoms that belong to the same molecule, i.e., $\mu_M = \max_l (\max_{i,j \in \Gamma_{\pi,l}, i \neq j} | \langle d_i, d_j \rangle |)$ and assume that $\mu_M \leq \frac{1}{n-1}$. Then, the distance between any molecule prototype $m_{\pi,l}$ and any of its realizations $m_{x,l}$ is bounded by

$$\|m_{x,l} - m_{\pi,l}\| \leq \sqrt{\frac{((1 + E)^2 \tan^2 \phi + E^2)n}{1 - (n - 1)\mu_M}}$$

Proof. For the molecule prototype $m_{\pi,l} = \sum_{i \in \Gamma_{\pi,l}} c_{\pi,l}(i) d_i$ a molecule realization can be written as :

$$m_{x,l} = \sum_{i \in \Gamma_{\pi,l}} v_i = \sum_{i \in \Gamma_{\pi,l}} (e_i d_i + p_i) = m_{\pi,l} + \sum_{i \in \Gamma_{\pi,l}} (p_i - [c_{\pi,l}(i) - e_i] d_i)$$

where an example of an approximation vector v_i for an atom d_i is show in Figure B.12. Therefore:

$$\|m_{x,l} - m_{\pi,l}\| = \left\| \sum_{i \in \Gamma_{\pi,l}} (p_i - [c_{\pi,l}(i) - e_i] d_i) \right\| \leq \sum_{i \in \Gamma_{\pi,l}} \|p_i - (c_{\pi,l}(i) - e_i) d_i\| \quad (\text{B.1})$$

by the triangle inequality. However, p_i is vertical to the direction of d_i . Therefore:

$$\|p_i - (c_{\pi,l}(i) - e_i) d_i\| = \sqrt{\|p_i\|^2 + \|(c_{\pi,l}(i) - e_i) d_i\|^2} = \sqrt{e_i^2 \tan^2 \phi_v + (c_{\pi,l}(i) - e_i)^2}$$

Substituting in Eq. (B.1), we get:

$$\|m_{x,l} - m_{\pi,l}\| \leq \sum_{i \in \Gamma_{\pi,l}} \sqrt{e_i^2 \tan^2 \phi_v + (c_{\pi,l}(i) - e_i)^2} \leq \sqrt{(1 + E)^2 \tan^2 \phi + E^2} \|c_{\pi,l}\|_1 \quad (\text{B.2})$$

since $|e_i| \leq E c_{\pi,l}(i), \forall l, i \in \Gamma_{\pi,l}$ and $c_{\pi,l}(i) \geq 0, \forall l, i$. For the $\|c_{\pi,l}\|_1$, given $\|c_{\pi,l}\|_1 \leq n$, we have :

$$\|c_{\pi,l}\|_1 \leq \|c_{\pi,l}\|_2 \sqrt{n} \quad (\text{B.3})$$

To bound the l_2 norm, we use the Rayleigh quotient $R(M, x) = \frac{x^T M x}{x^T x}$ and its bound $\lambda_{\min}(M) \leq R(M, x)$. In our case, $M = D_{\Gamma_{\pi,l}}^T D_{\Gamma_{\pi,l}}$ where $D_{\Gamma_{\pi,l}}$ is the matrix of the atoms participating in molecule $m_{\pi,l}$. Then, for $x = c_{\pi,l}$ we have :

$$\lambda_{\min}(D_{\Gamma_{\pi,l}}^T D_{\Gamma_{\pi,l}}) \leq \frac{1}{\|c_{\pi,l}\|_1^2} \Leftrightarrow \|c_{\pi,l}\|_1 \leq \frac{1}{\sqrt{\lambda_{\min}(D_{\Gamma_{\pi,l}}^T D_{\Gamma_{\pi,l}})}} \quad (\text{B.4})$$

where λ_{\min} is the minimum eigenvalue of $D_{\Gamma_{\pi,l}}^T D_{\Gamma_{\pi,l}}$. Finally, from the Gershgorin circle theorem applied on $D_{\Gamma_{\pi,l}}^T D_{\Gamma_{\pi,l}}$, which is the Gram matrix of $D_{\Gamma_{\pi,l}}$ and has as entries the inner products of the atoms in $\Gamma_{\pi,l}$, we get:

$$|\lambda - 1| \leq \max_{i \in \Gamma_{\pi,l}} \sum_{j \neq i, j \in \Gamma_{\pi,l}} |\langle d_i, d_j \rangle|$$

Since $\mu_M = \max_l (\max_{i,j \in \Gamma_{\pi,l}, i \neq j} | \langle d_i, d_j \rangle |)$ we get that $\forall l$:

$$1 - (n - 1)\mu_M \leq \lambda_{\min}(D_{\Gamma_{\pi,l}}^T D_{\Gamma_{\pi,l}})$$

Assuming $1 - (n - 1)\mu_M > 0 \Leftrightarrow \mu_M \leq \frac{1}{n-1}$ and substituting in Eq. (B.4) :

$$\|c_{\pi,l}\| \leq \frac{1}{\sqrt{1 - (n - 1)\mu_M}} \quad (\text{B.5})$$

Combining Eq. (B.3),(B.5) and (B.2) we finally get that :

$$\|m_{x,l} - m_{\pi,l}\| \leq \sqrt{\frac{((1 + E)^2 \tan^2 \phi + E^2)n}{1 - (n - 1)\mu_M}}$$

□

Now that we have established a bound for the distance $\|m_{x,l} - m_{\pi,l}\|$ between a molecule prototype and its realizations we can prove the following theorem providing a lower bound for the coherence μ_x of any dictionary DC_x with more than one realizations per prototype.

Theorem 2

With the distance between any molecule prototype and its realizations bounded by $\|m_{x,l} - m_{\pi,l}\| \leq r$ with $r < \frac{\sqrt{2}}{2}$, the coherence μ_x of any dictionary DC_x with more than one molecule realizations per molecule is

$$\mu_x \geq 1 - 2r^2 = L_x \quad (\text{B.6})$$

Proof. The coherence of a dictionary DC_x with more than one molecule realization per molecule is :

$$\mu_x = \max_{x,l,y,k} \frac{| \langle m_{x,l}, m_{y,k} \rangle |}{\|m_{x,l}\| * \|m_{y,k}\|} = \max_{x,l,y,k} | \cos \phi_{m_{x,l}, m_{y,l}} |$$

where $m_{x,l}, m_{y,l}$ are realizations of the molecules $m_{\pi,l}$ and $m_{\pi,k}$ respectively and $\phi_{m_{x,l}, m_{y,l}}$ is the angle between the two vectors. A lower bound to $\tilde{\mu}$ can be found by computing the maximum angle between two realizations of the same molecule, i.e. $l = k$. Then, $\mu_x \geq | \max_{x,y} \cos \phi_{m_{x,l}, m_{y,l}} |$.

Finally, we will use the same bound on the distance $\|m_{x,l} - m_{\pi,l}\|$ between a molecule prototype and its realizations to establish an upper bound for the coherence μ_u of any dictionary DC_x with more than one realizations per prototype. The following theorem provides the details.

Theorem 3

Let the coherence of the molecule prototype dictionary DC be μ . Given the bound on the distance between any molecule prototype and its realizations $\|m_{\pi,l} - m_{x,l}\| \leq r$ with $r < \frac{\sqrt{2}}{2}$, the coherence μ_u of any dictionary DC_x with at most one molecule realization per molecule is

$$\mu_u \leq U_u = \mu(1 - 2r^2) + 2r\sqrt{(1 - \mu^2)(1 - r^2)} \quad (\text{B.8})$$

Proof. We have:

$$\mu_u = \max_{x,y,l,k,l \neq k} \frac{|\langle m_{x,l}, m_{y,k} \rangle|}{\|m_{y,l}\| * \|m_{x,l}\|} = \max_{x,l,y,k} |\cos \phi_{m_{x,l}, m_{y,l}}| \quad (\text{B.9})$$

where $m_{x,l}, m_{y,l}$ are realizations of the molecules $m_{\pi,l}$ and $m_{\pi,k}$ respectively and $\phi_{m_{x,l}, m_{y,l}}$ is the angle between the two vectors. In the rest, we will restrict ourselves in the case where the angle $\phi_{m_{x,l}, m_{y,l}}$ that maximizes the Eq. (B.9) is less or equal to $\frac{\pi}{2}$. In the opposite case, a similar analysis can be followed and the final bound on μ_u is the same. Under this assumption,

$$\hat{\mu} = \max_{l,k,l \neq k} \cos \phi_{\tilde{m}_l, \tilde{m}_k} \quad (\text{B.10})$$

Moreover, we can assume that the indices l, k that maximize Eq. (B.10) are the same with the ones that maximize the equation $\mu = \max_{l,k} |\langle m_{\pi,l}, m_{\pi,k} \rangle| = \max_{l,k} \cos \phi_{m_{\pi,l}, m_{\pi,k}}$. In other words, we assume that the molecule prototypes that are the most coherent are also the ones that give rise to the most coherent realizations. Therefore, we will continue our analysis for the case where $\cos \phi_{m_{\pi,l}, m_{\pi,k}} = \mu$. Whats more, it is sufficient to restrict the rest of the analysis on the plane defined by the molecules prototypes $m_{\pi,l}, m_{\pi,k}$. This is possible because we model the space occupied by each prototype's realizations as a sphere, and as a result the plane defined by the centers of spheres is the space where the minimum distance and angle points between the spheres live.

The geometry on this plane is shown in Figure B.13. From the Figure we have that:

$$\phi \leq \phi_{m_{\pi,l}, m_{\pi,k}} \Leftrightarrow \phi_{lk} - 2\phi_S \leq \phi_{m_{x,l}, m_{y,k}} \Leftrightarrow \cos(\phi_{lk} - 2\phi_S) \geq \cos \phi_{m_{x,l}, m_{y,k}}$$

Therefore, using Eq. (B.10), we have:

$$\mu_u \leq \cos(\phi_{lk} - 2\phi_S) \quad (\text{B.11})$$

However, from trigonometry we have :

$$\cos(\phi_{lk} - 2\phi_S) = \cos \phi_{lk} \cos 2\phi_S + \sin \phi_{lk} \sin 2\phi_S \quad (\text{B.12})$$

Since $\cos \phi_{lk} = \mu$, we also have $\sin \phi_{lk} = \sqrt{1 - \cos^2 \phi_{lk}} = \sqrt{1 - \mu^2}$. Moreover from the triangle $OCm_{\pi,l}$ we have $\cos 2\phi_S = 1 - 2r^2$ and $\sin(2\phi_S) = \sqrt{1 - \cos^2(2\phi_S)} = \sqrt{1 - (1 - 2r^2)^2} = 2r\sqrt{1 - r^2}$. Substituting the above in Eq. (B.12) we get:

$$\cos(\phi_{lk} - 2\phi_S) = \mu(1 - 2r^2) + 2r\sqrt{(1 - \mu^2)(1 - r^2)}$$

Substituting this expression in Eq. (B.11), we get :

$$\mu_u \leq \mu(1 - 2r^2) + 2r\sqrt{(1 - \mu^2)(1 - r^2)}$$

□

- [1] B. A. Olshausen, D. J. Field, Sparse coding with an overcomplete basis set: a strategy employed by v1, *VisR* 37 (23) (1997) 3311–3325.
- [2] S. G. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, *IEEE Trans. on Signal Processing* 41 (12) (1993) 3397–3415.
- [3] J. A. Tropp, A. C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, *IEEE Trans. on Information Theory* 53 (12) (2007) 4655–4666.
- [4] R. Tibshirani, Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society* (1994) 267–288.
- [5] S. S. Chen, D. L. Donoho, M. A. Saunders, Atomic decomposition by basis pursuit, *SIAM Journal on Scientific Computing* 20 (1) (1998) 33–61.
- [6] M. Yuan, M. Yuan, Y. Lin, Y. Lin, Model selection and estimation in regression with grouped variables, *Journal of the Royal Statistical Society* 68 (1) (2006) 49–67.

- [7] P. Zhao, G. Rocha, B. Yu, The composite absolute penalties family for grouped and hierarchical variable selection, *Annals of Statistics* (2009) 3468–3497.
- [8] R. Jenatton, J.-Y. Audibert, F. Bach, Structured variable selection with sparsity-inducing norms, *The Journal of Machine Learning Research* 12 (2011) 2777–2824.
- [9] J. Huang, T. Zhang, D. Metaxas, Learning with structured sparsity, *The Journal of Machine Learning Research* 12 (2011) 3371–3412.
- [10] L. Jacob, G. Obozinski, J.-P. Vert, Group lasso with overlap and graph lasso, in: *Int. Conf. on Machine Learning (ICML)*, 2009, pp. 433–440.
- [11] S. Karygianni, P. Frossard, Structured sparse coding for image denoising or pattern detection, in: *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 3533–3537.
- [12] L. Daudet, Sparse and structured decompositions of signals with the molecular matching pursuit, *IEEE Trans. on Audio, Speech, and Language Processing* 14 (5) (2006) 1808–1816.
- [13] R. Rubinstein, M. Zibulevsky, M. Elad, Double sparsity: Learning sparse dictionaries for sparse signal approximation, *IEEE Trans. on Signal Processing* 58 (3) (2010) 1553–1564.
- [14] T. Peleg, Y. C. Eldar, M. Elad, Exploiting statistical dependencies in sparse representations for signal recovery, *IEEE Trans. on Signal Processing* 60 (5) (2012) 2286–2303.
- [15] P. J. Garrigues, B. A. Olshausen, Learning horizontal connections in a sparse coding model of natural images, in: *Advances in Neural Information Processing Systems (NIPS)*, 2008, pp. 505–512.
- [16] V. Cevher, M. F. Duarte, C. Hegde, R. G. Baraniuk, Sparse signal recovery using markov random fields, in: *Advances in Neural Information Processing Systems (NIPS)*, 2008, pp. 257–264.
- [17] K. Kavukcuoglu, M. Ranzato, R. Fergus, Y. LeCun, Learning invariant features through topographic filter maps, in: *IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 1605–1612.

- [18] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.
- [19] J. T. Rolfe, Y. LeCun, Discriminative recurrent sparse auto-encoders, arXiv preprint arXiv:1301.3775.
- [20] M. D. Zeiler, G. W. Taylor, R. Fergus, Adaptive deconvolutional networks for mid and high level feature learning, in: *IEEE Int. Conf. on Computer Vision (ICCV)*, 2011, pp. 2018–2025.
- [21] K. Jarrett, K. Kavukcuoglu, M. Ranzato, Y. LeCun, What is the best multi-stage architecture for object recognition?, in: *IEEE Int. Conf. on Computer Vision*, 2009, pp. 2146–2153.
- [22] Q. V. Le, Building high-level features using large scale unsupervised learning, in: *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 8595–8598.
- [23] H. Lee, R. Grosse, R. Ranganath, A. Y. Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, in: *Int. Conf. on Machine Learning (ICML)*, 2009, pp. 609–616.
- [24] J. Bruna, S. Mallat, Invariant scattering convolution networks, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 35 (8) (2013) 1872–1886.
- [25] T. T. Cai, L. Wang, G. Xu, Stable recovery of sparse signals and an oracle inequality, *IEEE Trans. on Information Theory* 56 (7) (2010) 3516–3522.
- [26] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM Journal on Imaging Sciences* 2 (1) (2009) 183–202.
- [27] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Foundations and Trends® in Machine Learning* 3 (1) (2011) 1–122.

- [28] E. J. Candes, M. B. Wakin, S. P. Boyd, Enhancing sparsity by reweighted l_1 minimization, *Journal of Fourier Analysis and Applications* 14 (5-6) (2008) 877–905.
- [29] Y. Lecun, C. Cortes, The mnist database of handwritten digits.
URL <http://yann.lecun.com/exdb/mnist/>
- [30] W. Dong, L. Zhang, G. Shi, X. Li, Nonlocally centralized sparse representation for image restoration., *IEEE Trans. on Image Processing* 22 (4) (2013) 1620–1630.
- [31] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Non-local sparse models for image restoration, in: *IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2009, pp. 2272–2279.