Chapter 2
The Fundamentals of Compressed Sensing

2.1 Sampling Theorems

Definition 2.1.1 Sampling is a fundamental way to represent and recover the continuous signals (analog domain) in the field of signal processing.

The Sampling theorem connects continuous signals and discrete signals. Figure.2.1 shows the procedure of the ideal sampling.

Theorem 2.1.1 (Nyquist Sampling Theorem [28]) *If a signal $x_a(t)$ is confined to be $[0, w_{max}]$ cycles per second, the signal can be reconstructed without loss by sampling it at more than $2w_{max}$ cycles per second as*

$$x_a(t) = \sum_{\infty}^{\infty} x(n) \frac{\sin \pi (2w_{max}t - n)}{\pi (2w_{max}t - n)} \tag{2.1}$$

where $x(n) = x_a(\frac{n}{2w_{max}})$.

For example, if one has a signal which is perfectly band limited to a band of f_0 within a time interval of T seconds, then one can reconstruct all the information in the signal by sampling it at discrete time as long as their sample rate, namely Nyquist rate, is greater than two times their bandwidth signal $(2f_0)$, known as Nyquist frequency. In case the bandlimit is too high or there is no bandlimit at all, the reproducing will derive imperfect result, named aliasing. Anyway, one can make an assumption that the signal has bandwidth B cps (cycle per second) with tiny values outside the interval T.

H. Cheng, *Sparse Representation, Modeling and Learning in Visual Recognition*,
Advances in Computer Vision and Pattern Recognition,
DOI 10.1007/978-1-4471-6714-3_2

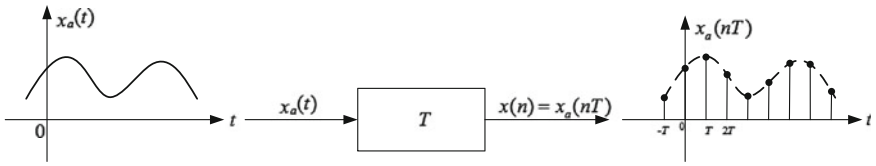

Fig. 2.1 The procedure of ideal sampling

Definition 2.1.2 (*Compressive Sampling*) If a signal $x_a(t)$ is compressible (i.e., K-sparse), we can have

$$x = \sum_{i=1}^{N} S_i \psi_i = \psi S \tag{2.2}$$

Thus, we can have compressive measurements via linear random procedure

$$y = Ax = A\psi S = \Theta S \tag{2.3}$$

where A or Θ is an $X \times N$ measurement matrix, a good measurement matrix can preserve the information in x, and one can recover x using various sparse recovery approaches in Chap. 3.

Compressive Sampling is to use linear random projection techniques to efficiently acquiring and reconstructing a compressible signal.

In general, the Nyquist-Shannon Sampling theorem only assumes the signal is band limited, and one can recover it without loss by sampling at $2w_{max}$. While compressive sampling depends on sparsity prior of the signal. Thus, Nyquist-Shannon sampling theorem ignores the sparsity prior. As a result, one has to increase Nyquist rate to guarantee the completion of the signals. This could be extremely worse in imaging systems (Fig. 2.2).

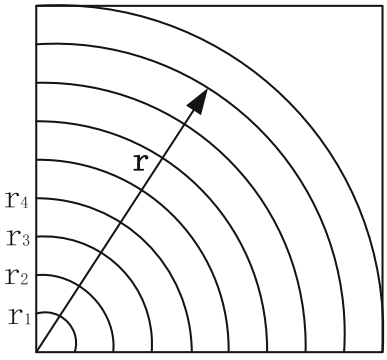

Fig. 2.2 Dimensionality reduction from 2D circles to 1D [28]

2.2 Compressive Sampling

As mentioned in previous section that Nyquist-Shannon sampling theorem is one of the tenet in signal precessing and information theory. It is worthwhile to restate the theorem here that: in order to perfectly reconstruct a signal, the number of samples needed is dictated by its bandwidth, i.e., the length of the shortest interval which contains the support, the spectrum of the signal. However, in the last few years, the "Compressive Sampling" has been the alternative theory and has emerged in the field of signal processing as well as information theory. The theory, basically, follows the concept of neural network in human brain which perceives information from outside world sparsely. With a small number of representation of the signal, human can most perfectly reconstruct the signal. Compressive sampling, similarly, shows that super-resolved signals and images can be reconstructed from far fewer data or measurements than what is usually considered important.

Compressive sampling has drawn much attention from research community in different fields ranging from statistics, information theory, coding theory, to theoretical computer science. We are going to summarize a few examples of compressive sampling in the real applications. JPEG2000 exploits the fact that many signals have a sparse representation, meaning that one can reconstruct, store, or transmit only a small numbers of adaptively chosen transform coefficients rather than all the signals samples. Another example can be illustrated in a digital camera. Instead of storing millions of imaging sensors, the pixels, it encodes the picture on just a few hundred kilobytes. In radiology, and biomedical imaging one is typically able to obtain far fewer data about an image of interest than the number of unknown pixels. Moreover, in wideband radio frequency signal analysis, a signal to be obtained at a rate which is much lower than the Nyquist rate because current Analog-to-Digital converter is limited. It is very obvious that typical signals have some structure; therefore, they can be compressed efficiently without much perceptual loss.

Mathematically, compressive sampling can be formulated as undersampled measurement problem as follows. Given a signal y_k, we aim to reconstruct a vector $x \in \mathbb{R}^N$ from linear measurements y as

$$y_k = \langle x, \varphi_k \rangle , \quad k = 1, \ldots, K, \quad or \quad y = \Phi x. \tag{2.4}$$

We are now trying to acquire information about the unknown signal by sensing x against K vectors $\varphi_k \in \mathbb{R}^N$. We are interested in the case that $K \ll N$ (underdetermined), where we have more unknown signal values than measurements. At first glance, the problem seems impossible. Sparse recovery techniques, discussed later in Chap. 3, will make the problem feasible.

2.2.1 Random Projection and Measurement Matrix

Johnson-Lindenstrauss lemma [23] has been a classic result of concerning low-distortion embeddings of points from high-dimensional into low-dimensional Euclidean space. The lemma is widely used in compressive sensing, manifold learning, dimensionality reduction, and graph embedding. Suppose we have n points $u_1, \ldots, u_n \in \mathbb{R}^d$, where d is large. We are going to map φ that $\mathbb{R}^d \longrightarrow \mathbb{R}^k$, where $k \ll d$, such that for each $i, j, 1 \leq i < j \leq n$, we have $(1-\varepsilon)||u_i - u_j||^2 \leq ||\varphi(u_i) - \varphi(u_j)||^2 \leq (1+\varepsilon)||u_i - u_j||^2$. The Johnson-Lindenstrauss lemma gives a randomized procedure to construct such a mapping with $k = O(\varepsilon^{-2} \log n)$. The embedding is a linear projection into a random k-dimensional subspace.

1. Random Projection

Random Projection is to use a random matrix $A \in \mathbb{R}^{n \times m}$ whose rows have unit length to project data from the high-dimensional data space $\boldsymbol{x} \in \mathbb{R}^m$ to a low-dimensional data space $\boldsymbol{v} \in \mathbb{R}^n$

$$\boldsymbol{v} = A\boldsymbol{x}, \tag{2.5}$$

where $n \ll m$. Ideally, we expect that A will provide a stable embedding that can preserve the distance between all pairs of original data in high-dimensional space to the embedded data points in the low-dimensional space. Luckily, the Johnson-Lindenstrauss lemma asserts that with high probability the distance between the points in a vector space is approximately preserved if they are projected onto a randomly selected subspace with suitably high dimension. Refer to [1] for details.

Baraniuk et al. [2] even proved that the random matrix A satisfying the the Johnson-Lindenstrauss lemma will also satisfy the restricted isometry property in compressive sensing. Thus, if the random matrix A in Eq. (2.5) satisfies the Johnson-Lindenstrauss lemma, we can recover x with minimum error from v with high probability if x is compressive such as audio or image. In other words, we can guarantee that v preserves most of the information which x possesses.

2. Random Sparse Measurement Matrix

Traditionally, we always use the random Gaussian matrix $A \in \mathbb{R}^{n \times m}$, where $a_{ij} \sim N(0, 1)$ as the measurement matrix which satisfy the RIP condition. However, the matrix of this type is dense; so the memory and computational loads are still large when m is large. Zhang et al. [33] proposed a sparse random measurement matrix which consumes little memory is consumed and can greatly reduce the computation in data projection. The entries of sparse random measurement matrix can be defined as

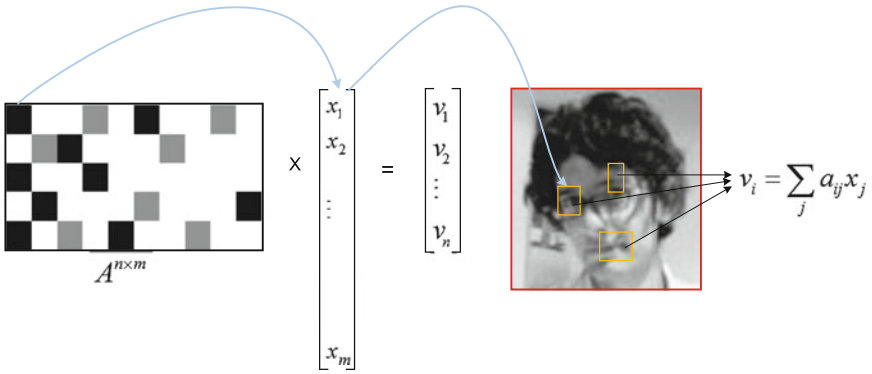

Fig. 2.3 Generating a compressive feature v from high-dimensional

$$a_{ij} = \sqrt{s} \times \begin{cases} 1 & \text{with probability } \frac{1}{2s} \\ 0 & \text{with probability } 1 - \frac{1}{s}. \\ -1 & \text{with probability } \frac{1}{2s} \end{cases} \tag{2.6}$$

Li et al. [25] showed that for $s = O(m)$ $(\boldsymbol{x} \in \mathbb{R}^m)$, then this matrix is asymptotically normal. Even when $s = m/\log(m)$, the random projections are almost as accurate as the conventional random projections where $a_{ij} \sim N(0, 1)$. In [33], they use $s = m/4$ to compress the data in visual tracking and get very good performance.

The process of data compressing by sparse random measure matrix is intuitively illustrated in Fig. 2.3, graphical representation of compressing a high-dimensional vector $\boldsymbol{x}$ to a low-dimensional vector $\boldsymbol{v}$. In the matrix A, dark, gray, and white rectangles represent negative, positive, and zero entries, respectively. The blue arrows illustrate that one of nonzero entries of one row of A sensing an element in $\boldsymbol{x}$ is equivalent to a rectangle filter convolving the intensity at a fixed position of an input image [33].

2.2.2 Sparsity

Sparsity, simply, means that the original signal is dense in a particular basis, however, after transformation into other convenient basis ψ, the coefficients under ψ offers a concise summary. Sparsity or compressibility has played and continue to play a fundamental and important role in many fields of science. Sparsity provides a solution in signal efficient estimations; for example, thresholding or shrinkage algorithms depend on sparsity to estimate the signal. Moreover, it leads to dimensionality reduction and efficient modeling. Sparsity even leads to signal compression where

the precision of a transform coders depends on the sparsity of the signal one wishes to decode.

The transformation from one basis to another can be viewed analytically as rotation of coordinate axes from the standard Euclidean basis to a new one. Why does it make sense to change coordinates in this way? Sparsity can provide the answer to such question. Take a look at several media types such as imagery, video, and acoustic, they all can be sparsely represented using transform-domain methods. For example, the media encoding standard JPEG is based on the notion of transform encoding, which means the data vector representing the raw pixels samples is transformed. Basically, JPEG relies on the discrete cosine transform (DCT)—a variant of the Fourier transform, while JPEG2000 based on the discrete wavelet transform (DWT).

The DCT of media content, technically, has transformed coefficients, which are quite large at the first several, but the rest are very small. Putting the those small coefficients to zeros and approximating the large coefficients by quantized representations will yield an approximate coefficient sequence which can be efficiently used to reconstruct the signal in a few bits. The approximate coefficient sequence can be inverse transformed to obtain an approximate representation of the original media content. On the other hand, the DWT has relatively few large coefficients, which are not necessarily at the first ones. Letting the small coefficients to zeros, and quantizing the large ones can obtain a sequence to be efficiently stored, and later inverse transformed to provide an approximate representation of the original media content. For many types of image content, JPEG2000 outperforms JPEG, while fewer bits are needed for a given accuracy or approximation. Thus, the success of DWT in image coding has close relationship to sparsity image content.

In short, sparsity of representation plays an important role in widely used techniques of transform-baseed image compression. It is also a driving factor for other important signal and image processing problems, including image denoising and image deblurring. Remarkably, it has been shown that a better representation is the one that is more sparse, i.e., less number of nonzero value.

2.2.3 Structured Sparsity

From the sparse representation research community point of view [30], sparsity has been roughly divided into two types. One is the pure or flat or unstructured sparsity which can be achieved by ℓ_0-norm, or ℓ_1-norm regularizer. Another is structured sparsity which usually can be obtained by different sparsity-inducing norms such as $\ell_{2,1}$-norm, $\ell_{\infty,1}$-norm, group ℓ_1-norm, and so on. In the flat sparsity or simply sparse representation, when regularizing with ℓ_0-norm, or ℓ_1-norm, each variable is selected individually, regardless of its position in the input feature vector, therefore, that existing relationships and structures between the variables, e.g., spatial, hierarchical or related to the physics of the problem at hand, are totally ignored. However, those

properties are very important or may improve the predictive performance in many applications.

Taking advantage from the prior knowledge has been shown effective in various applications. In neuroimaging based on functional magnetic resonance (fMRI) or magnetoencephalography (MEG), sets of voxels allowing to discriminate between different brain states are expected to form small localized and connected areas. Similarly, in face recognition, robust performance to occlusions can be improved by considering as features, sets of pixels that form small convex regions of the face which is beyond the capability of ℓ_1 regularization to encode such specific spatial constrains.

Such problems need the design of sparsity-inducing regularization schemes which are capable of encoding more sophisticated prior knowledge about the expected sparsity patterns.

2.3 ℓ_0, ℓ_1 and ℓ_2 Norms

In signal processing community, signals as real-valued functions which are divided into either continuous or discrete, and either infinite or finite. Norms play an important role on subspaces, and then we introduce the normed vector spaces. In this section, we shall consider the ℓ_0-norm, ℓ_1-norm, and ℓ_2-norm, and present the identities about them. First, we shall give the definition of the ℓ_p-norm

Definition 2.3.1 (ℓ_p*-norm*) $\boldsymbol{x}$ is a N-dimension vector, the ℓ_p-norm can be defined by the following formulation:

$$\|\boldsymbol{x}\|_{\ell_p} = \left(\sum_{i=1}^{N} |x_i|^p\right)^{\frac{1}{p}}. \tag{2.7}$$

By the definition of the ℓ_p-norm, we can easily define the ℓ_0-norm, ℓ_1-norm and ℓ_2-norm. When $p = 1$, we can define the ℓ_1-norm as follows:

$$\|\boldsymbol{x}\|_{\ell_p} = \begin{cases} \left(\sum_{i=1}^{N} |x_i|^p\right)^{\frac{1}{p}}, & p \in [1, \infty) \\ \max_{(i=1,2,\ldots,N)} |x_i|, & p = \infty \end{cases} \tag{2.8}$$

Note that the standard inner product in R^N leads to the ℓ_2-norm $\|\boldsymbol{x}\|_{\ell_2} = \sqrt{< x, x >}$. But, we should note that the ℓ_0-norm is not really a norm because it does not have some properties of the norm, it is defined as follows:

$$\|\boldsymbol{x}\|_{\ell_0} = |supp(x)| = \lim_{p \to 0} \|\boldsymbol{x}\|_p^p. \tag{2.9}$$

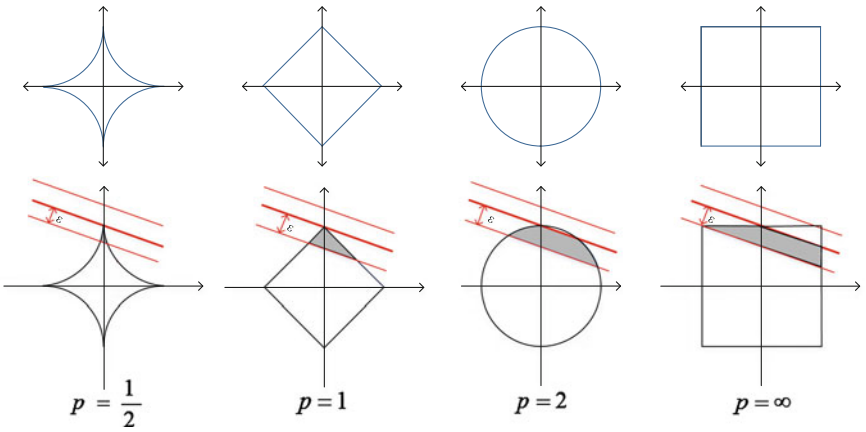

Fig. 2.4 The feasible region of $\|x\|_{\ell_p} = 1$ with $p = \frac{1}{2}$, 1, 2 and ∞, respectively

That means, the ℓ_0 is used to count the nonzero elements number of vector $\boldsymbol{x}$. The ℓ_p-norm with $p = \frac{1}{2}$, $p = 1$ and $p = 2$ can be seen intuitively in Fig. 2.4.

From the Fig. 2.4, we can see that when $0 < p < 1$ the ℓ_p-norm is not smooth and the vertex of the ℓ_p-norm at the coordinate axis; so it can be used as the regularization term to get the sparse solution. However it is nonconvex, so, it is not suitable to get the global optimal solution. When $p > 1$ the ℓ_p-norm is convex and smooth, it can be used as regularization term to get the global optimal solution. However, it is smooth and not suitable to get the sparse solution. Only in the situation $p = 1$, ℓ_p-norm both keep the convex and sparse properties; we would like to point out that not only the directive ℓ_1-norm is used to seek the sparse solution, but also the log-sum of ℓ_1-norm is used [31, 34]. Moreover, the log-sum of ℓ_1-norm is same as the ℓ_0 in principal. Shen et al. provided a rigorous justification for its optimization problem and the iterative reweighted method [31].

We can use norms to measure the residuals in many engineering problems. For example, we want to approximate $x \in R^N$ using $\widetilde{x} \in A$, a one-dimensional affine space. Then we can formulate the approximation error suing an ℓ_p-norm an $\text{argmin}_{x \in A} \|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\|_{\ell_p}$. From the Fig. 2.4. We can see that different norms results in different residuals. The procedure of seeking the optimal $\widetilde{x}$ is equivalent to grow on ℓ_p spark centered on x until it intersects with A. Note that laver p correspondents to more even residual among the coefficients while smaller p correspondents to more an evenly distributed of residuals. $\widetilde{x}$ is proved to sparse in latter case which has some amazing properties in high dimensions.

2.4 Spark and Singleton Bound

As we known from the Chap. 1, the sparse representation problem is used to solve the underdetermine problem $A\boldsymbol{x} = \boldsymbol{y}$. This problem has infinite solutions, so we need to add a prior to get the unique solution. The prior knowledge that we add is the assumption that solution is sparse, so we add a $\|\boldsymbol{x}\|_{\ell_0}$ regularization term to constrain the sparsity of solution. In order to study the uniqueness of solution, a key property we need to know is the *spark* of matrix A, which is mentioned in [15]. The rank of a matrix A is defined as the maximal number of columns A that are linearly independent, and its evaluation is a sequential process required L steps. However, calculation spark(A) quires a combinational process of 2^L steps.

Definition 2.4.1 (*Spark*) The *spark* of matrix A is the smallest possible number of its columns which are linearly dependent as

$$Spark(A) = \min_{\boldsymbol{x} \neq \boldsymbol{0}} \|\boldsymbol{x}\|_{\ell_0}, s.t. \quad A\boldsymbol{x} = \boldsymbol{0} \tag{2.10}$$

Definition 2.4.2 (*The Singleton Bound*) The highest spark of an matrix $A \in \mathbb{R}^{D \times N}$, $(D < N)$ is less than or equal to $D + 1$ [24].

The *spark* gives the simple criterion for the sparse solution of the $A\boldsymbol{x} = \boldsymbol{0}$ problem. The solution must satisfied that $\|\boldsymbol{x}\|_{\ell_0} \geq spark(A)$ to avoid trivial solutions. By the definition of spark, we can get the Theorem 2.4.1 as follows:

Theorem 2.4.1 (Uniqueness-Spark) *If a system of linear equations $A\boldsymbol{x} = \boldsymbol{y}$ has a solution $\boldsymbol{x}^*$ obeying $\|\boldsymbol{x}^*\|_{\ell_0} < spark(A)/2$, this solution is necessarily the sparsest possible [18].*

As we know the definition of *spark*, computing the spark of a matrix is a NP-hard problem, so, we need to find a simpler way to guarantee the uniqueness. The simplest way is to use the mutual-coherence of matrix A, which is defined as follows:

Lemma 2.4.1 (Mutual-Coherence) *The mutual-coherence of matrix A is the largest absolute normalized inner product between different columns from A. The mutual-coherence is defined as [18]*

$$\mu(A) = \max_{1 \leq i,j \leq N,\ i \neq j} \frac{|A_{.i}^T A_{.j}|}{\|A_{.i}\|_{\ell_2} \cdot \|A_{.j}\|_{\ell_2}}. \tag{2.11}$$

We can note that the mutual-coherence is used to characterize the dependence of the matrix columns. The mutual-coherence of the unitary matrix is zero. The $D \times N$ matrix A with $D < M$, and the mutual-coherence of this matrix is strictly positive, so we want the smallest value in order to make the matrix A as close as possible to the unitary matrix. So, we can use the mutual-coherence to find the lower bound of the spark, which is very hard to get.

Definition 2.4.3 (*Spark Lower Bound*) For any matrix $A \in \mathbb{R}^{D\times N}$, the following relationship holds [18]:

$$spark(A) \geq 1 + \frac{1}{\mu(A)} \tag{2.12}$$

By the Theorem 2.4.1, we get the uniqueness about mutual-coherence.

Definition 2.4.4 (*Uniqueness-Mutual-Coherence*) If a linear system of equations $A\boldsymbol{x} = \boldsymbol{y}$ has a solution $\boldsymbol{x}^*$ obeying $\|\boldsymbol{x}\|_{\ell_0} < \frac{1}{2}(1 + 1/\mu(A))$, this solution is necessarily the sparsest possible [18].

2.5 Null Space Property

The previous section tells us, if we want to recover a K-sparse signal, what condition about the spark and mutual-coherence of the sensing matrix we needed. In this section, we shall talk about the condition of sensing matrix from null space aspect, if we want to recover a K-sparse signal or approximately sparse signals. First, let us give the definition of the null space,

Definition 2.5.1 (*Null Space*) The null space of matrix A is denoted as follows:

$$\mathcal{N}(A) = \{\boldsymbol{z} : A\boldsymbol{z} = \boldsymbol{0}\}. \tag{2.13}$$

From the Theorem 2.4.1, we know $Spark(A) \in [2, D + 1]$. Therefore, it requires that $D \geq 2K$, if we want to recover a k-sparse signal. However, not all the signal is so sparse as we want. In some situations, we only have to deal with the approximately sparse signals, so we need to consider more restrictive conditions on the null space of A [11]. Suppose that $\Lambda \subset \{1, 2, \ldots, N\}$ is the subset of indices and $\Lambda^c = \{1, 2, \ldots, N\}/\Lambda$, $\boldsymbol{x}_\Lambda$ is the vector $\boldsymbol{x}$ with setting index Λ^c to be zero, $\sum_K = x_i$, $\|\boldsymbol{x}\|_0 \ll k$.

Definition 2.5.2 (*Null Space property*) A matrix A satisfies the **Null Space Property** (NSP) of order k if there exists a constant $C > 0$ such that [14],

$$\|h_\Lambda\|_{\ell_2} \leq C\frac{\|h_{\Lambda^c}\|_{\ell_1}}{\sqrt{K}} \tag{2.14}$$

holds for all $h \in \mathcal{N}(A)$ and for all Λ such that $|\Lambda| \leq K$.

From the definition of NSP, we can see that vectors in null space of A should not be concentrated on small subset of indices. By the NSP, we can give the conclusion about how to measure the performance of sparse recovery algorithm when dealing with general nonsparse signal $\boldsymbol{x}$. We define $\Delta : \mathbb{R}^D \to \mathbb{R}^N$ as the recovery algorithm,

$\Sigma_K = \{\boldsymbol{x} : \|\boldsymbol{x}\|_{\ell_0} \leq K\}$, and $\sigma_K(\boldsymbol{x})_p = \arg\min_{\hat{\boldsymbol{x}} \in \Sigma_K} \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_{\ell_p}$. The algorithm condition is as follows:

$$\|\Delta(A\boldsymbol{x}) - \boldsymbol{x}\|_{\ell_2} \leq C \frac{\sigma_K(\boldsymbol{x})_1}{\sqrt{K}}. \tag{2.15}$$

This algorithm can exactly recover the K-sparse signals, and also have a degree of robustness to non-sparse signals which depends on how well the signals are approximated by the K-sparse signals [11]. Then we shall give the theorem about NSP and the recovery algorithm.

Theorem 2.5.1 ([11]) *Let $A : \mathbb{R}^N \to \mathbb{R}^D$ denote a sensing matrix and the $\Delta : \mathbb{R}^D \to \mathbb{R}^N$ denote the arbitrary algorithm. If the pair (A, Δ) satisfies Eq. (2.15) then A satisfies the NSP of order $2K$.*

2.6 Uniform Uncertainty Principle, Incoherence Condition

In this section, we aim to introduce uncertainty principle proposed by Donoho and Stark [16, 17]. The UUP is a fundamental law in compressed sensing for signal representation and ℓ_1 uniqueness proof. First of all, the UUP is a fundamental law of signal resolution for sparse signal representation. In other word, we can represent a signal in a sparse way but it is strictly limited by this principle. Second, the NP is used to proof the ℓ_1 uniqueness.

From the previous section, NSP is a necessary condition for Eq. (2.15), which guarantees the algorithm can recover the K-sparse signal and ensures a degree of robustness to approximate the non-sparse signal. But NSP does not consider about the noisy situation. If the signals are contaminated, we should consider about other stronger conditions. Candes and Tao [7] had introduced the uniform uncertainty principle (UUP). It aims to define the "Restricted Isometry Property (RIP)" of the sensing matrix A, which plays a very important role in compressed sensing. First, we need to know what is the *Uncertainty Principle* which is given as follows:

Theorem 2.6.1 (Uncertainty Principle [16]) *The time domain signal $\boldsymbol{y} \in \mathbb{R}^D$ has the sparsity K_t under the transformation $\boldsymbol{y} = A_t \boldsymbol{x}_t$, where A_t is the identity matrix, and the Fourier transform $\overline{\boldsymbol{y}} \in \mathbb{R}^D$ has the sparsity K_w under the transformation $\overline{\boldsymbol{y}} = A_w \boldsymbol{x}_w$. Then, the two sparsity parameters should satisfy*

$$K_t K_w \geq D \geq \frac{1}{\mu^2}, \tag{2.16}$$

and

$$K_t + K_w \geq 2\sqrt{D} \geq \frac{2}{\mu}, \tag{2.17}$$

where μ is the maximum correlation of the two bases A_t and A_w which is defined as

$$\mu := \max_{i,j} \left\{ |\langle A_{t(.i)} A_{w(.j)} \rangle| \right\} \tag{2.18}$$

The *Uncertainty Principle* tell us that any signal cannot be sparsely represented in both domains simultaneously. If the sparsity in one domain is fixed, i.e., K_t, then, the sparsity level obtainable in the other domain shall be limited, i.e., $K_w \geq \frac{2}{\mu} - K_t$. By the uncertainty principle, Candes et al. propose the *Uniform Uncertainty Principle* which is the fundamental knowledge of compressed sensing.

Definition 2.6.1 (*Uniform Uncertainty Principle (UUP)* [8]) We can say a measurement matrix or sensing matrix A satisfy the *Uniform Uncertainty Principle* with oversampling factor λ if for every sufficiently small $\alpha > 0$, the following statement is true with probability at least $1 - O(N^{-\rho/\alpha})$ for some fixed positive constant $\rho > 0$: for all subsets T such that

$$|T| \leq \alpha \cdot K/\lambda, \tag{2.19}$$

the sensing matrix A obeys the bounds

$$\frac{1}{2} \cdot \frac{K}{N} \|\boldsymbol{x}\|_{\ell_2}^2 \leq \|A\boldsymbol{x}\|_{\ell_2}^2 \leq \frac{3}{2} \frac{K}{N} \|\boldsymbol{x}\|_{\ell_2}^2, \tag{2.20}$$

holding for all signals $\boldsymbol{x}$ with support size less or equal to $\alpha K/\lambda$.

Definition 2.6.2 A matrix A satisfies the Restricted Isometry Property of order K if there exists a $\delta_K \in (0, 1)$ such that

$$(1 - \delta_K)\|\boldsymbol{x}\|_{\ell_2}^2 \leq \|A\boldsymbol{x}\|_{\ell_2}^2 \leq (1 + \delta_K)\|\boldsymbol{x}\|_{\ell_2}^2, \tag{2.21}$$

holds for all K-sparse vector $\boldsymbol{x}$, where δ_K is called K-restricted isometry constants.

The RIP condition provides the basic condition for the compressed sensing theory. If the matrices satisfies the RIP condition, many good things are guaranteed such as the ℓ_1 recovery is equivalence with ℓ_0 recovery. If a matrix A satisfies the RIP of order $2K$, then the Eq. (2.21) can be interpreted as the matrix A approximately preserves the distance between any pair of K-sparse vector, thus we can use this matrix to recover the K-sparse signal. We can note from the definition of RIP condition that if matrix A satisfies the RIP of order K with constant δ_K, then for any order $K' < K$ the matrix A also satisfies the RIP of order K' with constant $\delta_{K'}$. Needell and Tropp [27] also present that if the sensing matrix A satisfies the RIP with a very small constant, then the matrix A also satisfies RIP of order γK for certain γ, with a worse constant. This property is presented as follows:

Lemma 2.6.1 *Suppose that A satisfies the RIP of order K with constant δ_K. Let γ be a positive integer. Then A satisfies the RIP of order $K' = \gamma \lfloor \frac{K}{2} \rfloor$ with constant $\delta_{K'} < \gamma \cdot \delta_K$, where $\lfloor \cdot \rfloor$ denotes the floor operator.*

In many cases, the signal is non-sparse but it can be represented as sparse signal under some specific orthogonal basis; for example, $\boldsymbol{s} = \Psi \boldsymbol{x}$, where, $\boldsymbol{s}$ is non-sparse signal and Ψ is the orthogonal basis, $\Psi^T \Psi = \Psi \Psi^T = I_N$. The $D \times 1$ measurement vector $\boldsymbol{y}$ can be expressed as

$$\boldsymbol{y} = \Phi \boldsymbol{s} = \Phi \Psi \boldsymbol{x} := A\boldsymbol{x}, \tag{2.22}$$

where Φ is $D \times N$ sensing matrix.

Definition 2.6.3 (*Incoherence Condition*) The Incoherence Condition can be defined as the rows of Φ should be incoherent to the columns of Ψ.

We can note that if Φ and Ψ could not satisfy the incoherence condition, for example, in the extreme case, selecting the first D column of Ψ as the D rows of Phi we can get

$$\Phi \Psi = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}. \tag{2.23}$$

We can easily find that this matrix $\Phi \Psi$ can never satisfy the RIP condition. We can use the i.i.d. Gaussian to construct the sensing matrix which has been proved that it will be in coherent to any basis.

2.7 ℓ_1 and ℓ_0 Equivalence

The RIP [7] is used to prove the equivalence between ℓ_1 and ℓ_0-norm in sparse signal recovery. It has been proved in [7] that the solution of dual ℓ_0 minimization problems

(1) Sparse Error Correction: Given $\boldsymbol{y} \in \mathbb{R}^N$, $A \in \mathbb{R}^{N \times M}$ $(N > M)$,

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - A\boldsymbol{x}\|_{\ell_0}, \tag{2.24}$$

(2) Sparse Signal Reconstruction: Given $\boldsymbol{z} \in \mathbb{R}^D$, $B \in \mathbb{R}^{D \times N}$ $(D < N)$,

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \|\boldsymbol{w}\|_{\ell_0} \quad \text{s.t.} \quad \boldsymbol{z} = B\boldsymbol{w}, \tag{2.25}$$

are the same as the solutions of problems

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - A\boldsymbol{x}\|_{\ell_1}, \tag{2.26}$$

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \|\boldsymbol{w}\|_{\ell_1} \quad \text{s.t.} \quad \boldsymbol{z} = B\boldsymbol{w}, \tag{2.27}$$

if the error $\boldsymbol{e} = \boldsymbol{y} - A\boldsymbol{x}$ or the solution $\boldsymbol{w}$ is sufficiently sparse. Y. Sharon et al. [29] verify the equivalence between ℓ_1 and ℓ_0 minimization problem by the algorithm. First, they give the definition of "*d-skeleton*" as follows:

Definition 2.7.1 (*d-Skeleton* [29]) The "d-skeleton" is defined as the collection of all the d-dimensional faces of the standard ℓ_1-ball $B_1 \doteq \{\boldsymbol{v} \in \mathbb{R}^m : \|\boldsymbol{v}\|_{\ell_1} \leq 1\}$. We can denote $SK_d(B_1)$:

$$SK_d(B_1) \doteq \{\boldsymbol{v} \in \mathbb{R}^N : \|\boldsymbol{v}\| = 1, \|\boldsymbol{v}\|_{\ell_0} \leq d + 1\}. \tag{2.28}$$

By the definition of d-skeleton, they can prove the proposition as follows:

Proposition 2.7.1 *For every $\boldsymbol{x}_0 \in \mathbb{R}^M$ and $\boldsymbol{y} \in \mathbb{R}^N$, the following implication holds*

$$\|\boldsymbol{y} - A\boldsymbol{x}_0\|_{\ell_0} \leq T \quad \Rightarrow \quad \boldsymbol{x}_0 = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - A\boldsymbol{x}\|_{\ell_1} \tag{2.29}$$

if and only if

$$\forall \boldsymbol{v} \in SK_{T-1}(B_1), \quad \forall \boldsymbol{z} \in \mathbb{R}^M \backslash \boldsymbol{0}, \quad \|\boldsymbol{v} + A\boldsymbol{z}\|_{\ell_1} > 1. \tag{2.30}$$

We can note that the Eq. (2.29) is what we needed. But this proposition asks us to check starting from $T = 1, 2, \ldots$ until the condition (2.30) eventually fails. Even more, it asks us to check every point on the d-skeleton. Then, they propose the proposition which tells us an equivalent condition that does not require search over $\boldsymbol{v}$, and only involves checking a finite set of points in $span(A)$.

Proposition 2.7.2 *Let $A \in \mathbb{R}^{N \times M}$ and $d \in \mathbb{N} \cup 0$ be given and assume the rows of A are in general directions, i.e., any M rows of A are independent. The following holds:*

$$\forall \boldsymbol{v} \in SK_d(B_1), \quad \forall \boldsymbol{z} \in \mathbb{R}^M \backslash 0 \quad \|\boldsymbol{v} + A\boldsymbol{z}\|_{\ell_1} > 1 \tag{2.31}$$

if and only if for all subsets $I \subset Q \doteq \{1, \ldots, N\}$ containing $M - 1$ indices, all subsets $J \subset Q \backslash I$ containing $T = d + 1$ indices, and for some $\boldsymbol{y} \in \mathbb{R}^N$ such that

$$\boldsymbol{y} \in span(A) \backslash 0, \quad \forall i \in I \quad y_i = 0, \tag{2.32}$$

the following holds:

$$\sum_{j \in J} |y_j| < \sum_{j \in Q \backslash J} |y_j|. \tag{2.33}$$

2.8 Stable Recovery Property

In this section, we shall give some conclusions which can present RIP condition's necessity for signal $\boldsymbol{x}$ recovery from the measurements $A\boldsymbol{x}$, and even more it is necessary for stable recovery in case of noise [13]. Stable recovery stems from two issues. First of all, signals are not strictly sparse in practice. The small portion of the signal has large magnitude while the rest are close to zero but not exactly zero. Thus, there exits model error in a sparse model. Second, there always exit noise in the signal measured from sensors.

Definition 2.8.1 Let $A : \mathbb{R}^N \to \mathbb{R}^D$ denotes the sensing matrix and $\Delta : \mathbb{R}^D \to \mathbb{R}^N$ denotes the recovery algorithm. We say that the pair (A, Δ) is C-stable if $\forall \boldsymbol{x} \in \Sigma_k$ and $\forall \boldsymbol{e} \in \mathbb{R}^D$, we have that

$$\|\Delta(A\boldsymbol{x} + \boldsymbol{e}) - \boldsymbol{x}\|_{\ell_2} \leq C\|\boldsymbol{e}\|_{\ell_2}. \tag{2.34}$$

This definition tells us that if the measurements add some small amount of noise, the impact on the recovered signal should not be arbitrarily large. Next, we shall give a theorem which demonstrates that any recovery algorithm can stably recover the signal from noisy measurements requires that A satisfy the lower bound of RIP condition with a constant determined by C [13].

Theorem 2.8.1 *If the pair (A, Δ) is C-stable, then*

$$\frac{1}{C}\|\boldsymbol{x}\|_{\ell_2} \leq \|A\boldsymbol{x}\|_{\ell_2} \tag{2.35}$$

for all $\boldsymbol{x} \in \Sigma_{2K}$.

We can note that when $C \to 1$, the sensing matrix A can satisfy the lower bound of RIP condition with $\delta_K = 1 - 1/C^2 \to 0$. Thus, if we want to reduce the impact of the noise in the recovery algorithm, we must let sensing matrix A to satisfy the lower bound of RIP condition with a smaller δ_K.

Another aspect we need to consider is the dimension of the measures, and how many measurements are necessary to achieve the RIP. Now, we ignore the impact of the δ and only focus on the dimensions of the problem (D, N and k) then we can get a simple lower bound, which is proven in [13].

Theorem 2.8.2 *Let A be an $D \times N$ matrix that satisfies the RIP of order $2K$ with constant $\delta \in (0, \frac{1}{2}]$. Then*

$$D \geq CK \log\left(\frac{N}{K}\right), \tag{2.36}$$

where $C \approx 0.28$.

1. The Relationship Between the RIP and the NSP

Finally, we shall give the conclusion that when the matrix satisfies the RIP, it also satisfies the NSP, in other words, the RIP is strictly stronger than the NSP.

Theorem 2.8.3 *Suppose that sensing matrix A satisfies the RIP of order $2K$ with $\delta_{2K} < \sqrt{2} - 1$. Then A satisfies the NSP of order $2K$ with constant*

$$C = \frac{\sqrt{2}\delta_{2K}}{1 - (1 + \sqrt{2})\delta_{2K}}. \tag{2.37}$$

The prove detail of this theorem can be found in [19].

2. Signal Recovery via ℓ_0 and ℓ_1 Minimization

Let us consider the original problem that we want to solve the linear underdetermined problem $\boldsymbol{y} = A\boldsymbol{x}$ with constraint $\boldsymbol{x}$ as sparse as possible. The problem can be naturally solved by the following optimal equation:

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_{\ell_0} \quad \text{s.t.} \quad \boldsymbol{y} = A\boldsymbol{x}. \tag{2.38}$$

The performance of above method can be analyzed in [10, 22] which is under the appropriate assumptions on A, but we still do not have a sufficient method to solve this problem, because $\|\cdot\|_{\ell_0}$ is nonconvex and minimizes $\|\boldsymbol{x}\|_{\ell_0}$ is a NP-hard problem [26].

One of the tractable method is approximate ℓ_0-norm by ℓ_1-norm which preserves the sparsity and convex properties, and the reason can refer to Sect. 2.3.

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_{\ell_1} \quad \text{s.t.} \quad \boldsymbol{y} = A\boldsymbol{x}. \tag{2.39}$$

So, this problem is computational tractable and can be treated as a linear programing problem [9]. The following theorem is very remarkable. It consider about the case that $x \in \Sigma_K$ and if the sensing matrix A satisfies the RIP condition which only need $O(k \log(N/K))$ measurements, we can recover the K-sparse signal exactly.

If we only consider about the noise free case, we can get the following Lemma and Theorem.

Lemma 2.8.1 *Suppose that A satisfies the RIP of order $2K$, and let $h \in \mathbb{R}^N$, $h \neq 0$ be arbitrary. Let Λ_0 be any subset of $\{1, 2, \ldots, N\}$ such that $|\Lambda_0\| \leq K$. Define Λ_1 as the index set corresponding to the K entries of $h_{\Lambda_0^c}$ with largest magnitude, and set $\Lambda = \Lambda_0 \bigcup \Lambda_1$. Then*

$$\|h_\Lambda\|_{\ell_2} \leq \alpha \frac{\|h_{\Lambda_0^c}\|_{\ell_1}}{\sqrt{K}} + \beta \frac{|\langle Ah_\Lambda, Ah\rangle|}{\|h_\Lambda\|_{\ell_2}}, \tag{2.40}$$

where

$$\alpha = \frac{\sqrt{2}\delta_{2K}}{1-\delta_{2K}}, \quad \beta = \frac{1}{1-\delta_{2K}}. \tag{2.41}$$

By the Lemma 2.8.1 we can get the Lemma 2.8.2.

Lemma 2.8.2 *Suppose that A satisfies the RIP of order 2K with* $\delta_{2K} < \sqrt{2}-1$. *Let* $\boldsymbol{x}, \hat{\boldsymbol{x}} \in \mathbb{R}^N$ *be given, and define* $h = \hat{\boldsymbol{x}} - \boldsymbol{x}$. *Denote* Λ_0 *as the index set corresponding to the K entries of* $\boldsymbol{x}$ *with largest magnitude and* Λ_1 *the index set corresponding to the K entries of* $h_{\Lambda_0^c}$ *with largest magnitude. Set* $\Lambda = \Lambda_0 \bigcup \Lambda_1$. *If* $\|\hat{\boldsymbol{x}}\|_{\ell_1} \leq \|\boldsymbol{x}\|_{\ell_1}$, *then*

$$\|h\|_{\ell_2} \leq C_0 \frac{\sigma_k(\boldsymbol{x})_1}{\sqrt{k}} + C_1 \frac{|\langle Ah_\Lambda, Ah\rangle|}{\|h_\Lambda\|_{\ell_2}}, \tag{2.42}$$

where

$$C_0 = 2\frac{1-\left(1-\sqrt{2}\right)\delta_{2K}}{1-\left(1+\sqrt{2}\right)\delta_{2K}}, \quad C_1 = \frac{2}{1-\left(1+\sqrt{2}\right)\delta_{2K}}. \tag{2.43}$$

The Lemma 2.8.2 shows us that if the sensing matrix satisfies the RIP, the error bound of the general ℓ_1 minimization algorithm. If we consider about the problem of Eq. (2.39), the specific bounds is given by Theorem 2.8.4 as follows:

Theorem 2.8.4 *Suppose that sensing matrix A satisfies the RIP of order* $2K$ *with* $\delta_{2K} < \sqrt{2}-1$. *Then the solution* $\boldsymbol{x}^*$ *of problem of Eq. (2.39) obeys*

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_{\ell_2} \leq C_0 \frac{\sigma_K(\boldsymbol{x})_1}{\sqrt{K}}. \tag{2.44}$$

Right now, we consider the noisy case, because in the real-world systems the measurements always contaminated by some form of noise. We consider the worst situation to uniformly bound the noise [6].

Theorem 2.8.5 ([5]) *Suppose that A satisfies the RIP of order* $2K$ *with* $\delta_{2K} < \sqrt{2}-1$ *and let* $\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{e}$ *where* $\|\boldsymbol{e}\|_{\ell_2} \leq \varepsilon$. *The solution* $\boldsymbol{x}*$ *to Eq. (2.39) obeys*

$$\|\boldsymbol{x}^* - \boldsymbol{x}\|_{\ell_2} \leq C_0 \frac{\sigma_K(\boldsymbol{x})_1}{\sqrt{K}} + C_2\varepsilon, \tag{2.45}$$

where

$$C_0 = 2\frac{1-\left(1-\sqrt{2}\right)\delta_{2K}}{1-\left(1+\sqrt{2}\right)\delta_{2K}}, \quad C_1 = 4\frac{\sqrt{1+\delta_{2K}}}{1-\left(1+\sqrt{2}\right)\delta_{2K}}. \tag{2.46}$$

This theorem tells us that even in the noisy case, if the sensing matrix satisfies the RIP condition, the ℓ_1 algorithm also can recover the signal stably.

2.9 Information Theory

Information theory is an interdisciplinary branch of applied mathematics, computer science, and electrical engineering. The field was remarkably developed by Claude E. Shannon in his theorem in finding fundamental limits on signal processing operations, e.g., compressing, storing and communicating data. Since then it has been emerged in many applications such as natural language processing, cryptography, biology, statistical inference, information retrieval, and so on. Information theorem, mathematically, based on statistics and probabilistic theory.

2.9.1 K-sparse Signal Model

Definition 2.9.1 Signal modeling is the process of representing signals, with respect to an underlying structure of the signal's fundamental behavior, by some set of parameters.

Signal modeling is used in many applications including signal compression, prediction, reconstruction, and understanding. In signal compressive applications, rather than storing original signal, one needs to store a set of parameters, whose sizes are much smaller than the original signals, and can be used to reconstruct the original signal or at least as close as possible to the original signal. Usually, to reconstruct the signal, one needs to design a filter as shown in Fig. 2.5. In other words, the problem can be simply defined as given observations $X[n], n = 0, \ldots, N-1$ and filter order m and n, find the parameters of $H(z)$ such that the modeled output, $y[n] = h[n]$.

In the field of digital signal processing, the input–output relation of a linear time-invariant (LTI) system is given in the z domain by

$$Y(z) = \frac{B(z)}{A(z)} X(z) = H(z)X(z) \tag{2.47}$$

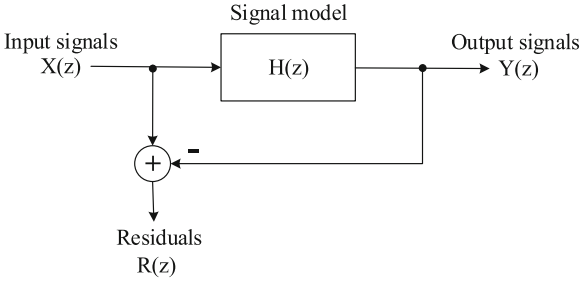

Fig. 2.5 A basic model for signal conversion

where $H(z)$ is filter response, $A(z)$ and $B(z)$ are polynomials. Therefore, $H(z)$ can be relaxed to

$$H(z) = \frac{\sum_{i=1}^{n} b_n z^{-n}}{\sum_{j=1}^{m} b_m z^{-m}} \tag{2.48}$$

Various models have been proposed in the literatures, e.g., AutoRegressive (AR), Moving Average (MA), AutoregRessive-Moving Average (ARMA), and low-rank or sparse model which is the main topic of the book.

We are now interested in vector x with K nonzero entries, i.e., K-sparse vector. We denote the indices of those nonzero entries of the vector x by $t = (t_1, t_2, \ldots, t_K)$, and name it as index profile. Moreover, each entry $t_k \in \{1, 2, \ldots, N\}$ denotes the index of a nonzero entry in x. Let S_t be the set of all feasible index profile. Its size can be defined as

$$|S_t| = \binom{N}{k} \tag{2.49}$$

We set the values of the K nonzero entries into a vector $s = (s_1, s_2, \ldots, s_K)$, and name it value profile which can be determined from a distribution. For instance, we could use Gaussian, Bernoulli, or a hybrid distribution. We use a p.d.f $f_s(s)$ to denote a VP distribution. For an example of complex valued Gaussian multivariate random vector, the p.d.f is obtained by

$$f_s(s) = \frac{1}{\pi^N |C_s|} \exp\left[-\frac{1}{2}(s - \bar{s})^* C_s^{-1} (s - \bar{s})\right] \tag{2.50}$$

where $\bar{s} := E\{s\}$ is the mean vector of the Gaussian multivariate s and $C_s := E\{(s - \bar{s})(s - \bar{s})\}$ is the covariance matrix.

In the case that support set size of vector x is smaller than or equal to K, the hybrid distribution should be used to overcome such problem. The index profile set S_t should include all feasible index profile whose size is smaller than or equal to K. Therefore, the size of the index profile set is equal to the number of points in a Hamming sphere of size K as

$$|S_t| = V_2(N, K) = \sum_{k=0}^{K} \binom{N}{k}. \tag{2.51}$$

Finally, we obtain the number of nonzero entries, k as a random variable with the following distribution

$$f_K(k) = \frac{\binom{N}{k}}{V_2(N, K)}. \tag{2.52}$$

To obtain a hybrid distribution, one could use the two distribution from Eqs. (2.49) to (2.51).

2.9.2 The Entropy of K-sparse Signals

In the field of information theory, entropy is the average amount of information contained in a message, i.e., events, samples, or characters drawn from a distribution or data stream. Entropy of a discrete variable X is a measure of the amount of uncertainty associated with the value of X.

Definition 2.9.2 The entropy $H(X)$ of a discrete random variable X is defined by

$$H(X) = -\sum_{x \in X} px(x) \log p(x). \tag{2.53}$$

The entropy is expressed in bits, therefore, the log is to the base 2. In case the logarithm is based in p, the entropy is defined as $H_b(X)$. Entropy does not depend on the value of random variable X, but depends on its probabilities. As entropy is a measure of unpredictability of information content, let us have a look at an example to get more intuitive understanding about it.

Now consider a coin-tossing problem. If the coin is fair which means when the probability of heads is the same as the probability of tails, then the entropy of the coin toss is as high as it could be. Obviously, there is no way to predict the outcome of the coin toss ahead of time? the best, thus we can do is to predict that the coin will come up with heads or tails, and our prediction will be correct with the probability 1/2. Such a coin toss has one bit of entropy since there are two possible outcomes that occur with equal probability, and learning the actual outcome contains one bit of information. Contrarily, a coin toss with a coin that has two heads and no tails has zero entropy since the coin will always come up heads, and the outcome can be predicted perfectly.

In previous subsection, we have discussed the fundamental concept of entropy. This subsection will illustrate the information in terms of bits which can be represented by the K-sparse signal x. In other words, we are interested in determining how large the entropy of the K-sparse signal x is. In general, K-sparse signal x has K nonzero entries. For example, given a signal vector $y_{M\times 1}$, we have to compute how much information in terms of bits which $y_{M\times 1}$ will represent. In order to answer this question, we shall divide the case into two exclusive ones. To simplify the answer, let denote D be an $M \times N$ Fourier transform matrix or dictionary of atoms with prime N. Thus, we obtain $y_{M\times 1} = D_{M\times N}x$. If the map is one-to-one correspondent, which means $M \geq 2K$, the entropy of x is the entropy of $y_{M\times 1}$.

Lemma 2.9.1 *Let D be an $M \times N$ Fourier transform matrix with prime N, where $M \geq 2K$. Let x be a K-sparse signal. Then, the entropy of y given D is $H(x)$, i.e., $H(y|D) = H(Dx|D) = H(x)$. If $M < 2K$, then $H(y|D) = H(Dx|D) \leq H(x)$.*

$$\begin{aligned} H(x) &= H(t = (t_1, \ldots, t_K), s = (s_1, \ldots, s_K)) \\ &= H(t) + H(s|t) \\ &= H(t) + H(s) \end{aligned} \tag{2.54}$$

Suppose that the supporting set of size K is uniformly randomly distributed, the entropy of $H(t_1, \ldots, t_K)$ can be written as

$$H(t_1, \ldots, t_K) = \log \binom{N}{K}. \tag{2.55}$$

Applying the Stirling's approximation for the factorial function, we can obtain

$$\log_2 \left(\frac{1}{N+1}\right) + NH\left(\frac{K}{N}\right) \leq \log_2 \binom{N}{k} \leq NH\left(\frac{N}{k}\right). \tag{2.56}$$

When N is large, one can derive $\log_2 \binom{N}{k} \cong NH\left(\frac{K}{N}\right)$. On the other hand, when K is small compared to N, the entropy function $H\left(\frac{K}{N}\right) = \frac{K}{N}\log_2 \frac{N}{K} + \left(\frac{N-K}{N}\right)\log_2 \left(\frac{N}{N-K}\right)$ can be approximated with the first term only, which means $H\left(\frac{K}{N}\right) \approx \frac{1}{N} K \log_2 \frac{N}{K}$. Thus, we can obtain when $K \ll N$, that

$$NH\left(\frac{K}{N}\right) \approx K \log_2 \left(\frac{N}{K}\right). \tag{2.57}$$

In conclusion, if $M \geq 2K$, any compression map from x to $y = Dx$ is one-to-one correspondent. Therefore, the entropy of y is also the same as the entropy of x.

2.9.3 Mutual Information

Mutual information of two variables X and Y can be defined as

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \tag{2.58}$$

where $p(x, y)$ is the joint probability distribution function of X and Y, and $p(x)$ and $p(y)$ are marginal probability distribution functions of X and Y, respectively. Obviously, the mutual information measures the mutual independence of variables, i.e., it measures how much information the variables share each other with the most common unit of the measurement bits. Mutual information, in information theory, can reduce the uncertainty of one variable given knowledge of another. From Eq. (2.58), one can see that if the two variable X and Y are completely independent, then $p(x, y) = p(x)p(y)$. Therefore,

$$\log \left(\frac{p(x, y)}{p(x)p(y)}\right) = \log 1 = 0 \tag{2.59}$$

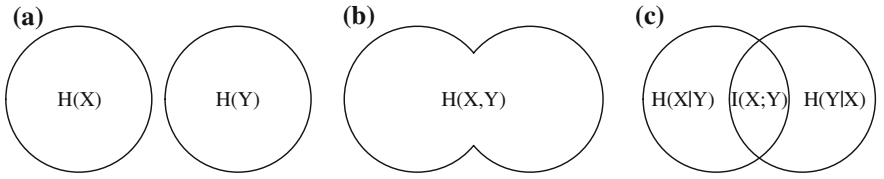

Fig. 2.6 The relationship between entropy and mutual information **a** Marginal entropies. **b** Join entropy. **c** Mutual information

which means X and Y do not share any information at all. The mutual information can be equivalently written as

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X|Y) \\ &= H(X,Y) - H(X|Y) - H(Y|X) \end{aligned} \tag{2.60}$$

where $H(X)$ and $H(Y)$ are the marginal entropies, and $H(X, Y)$ is the joint entropies of X and Y as shown in Fig. 2.6. If $H(X)$ is considered as a measure of uncertainty of variable, then $H(X|Y)$ is a measure of what Y does not say about X.

In communication channel, input X getting through transmission medium will produce output Y. In the perfect transmission, i.e., if the channel is noiseless, the input is equal the to output, namely $X = Y$. However, in real-world channels, the transmission medium is noisy, an input X is converted to an output Y with probability $P(Y|X)$.

Given a communication channel, for example, one can transmit any messages s from a set of M possible messages by performing following three steps.

(1) Assign a string $x = (x_1, x_2, \ldots, x_n)$ of length n to each message s. Each $x(s)$ is called *codeword*. The processing of generating from a set of message to a set of codeword is called encoding.
(2) Transmit the corresponding string $x(s)$ over the channel which yield output y with the same length n.
(3) Use output y to reconstruct the transmitted message x by using a deterministic function named *decoding*. The decoding then maps each y to one symbol s'.

Remarkably, first the number of transmitted messages is much less than the number of possible messages. Then, each message x_i is randomly and independently selected from a distribution, denoted as $P(X)$. Therefore, when one designs a communication channel, only M and $P(X)$ can be controlled. Thus, in general, one adjusts $P(X)$ to make the number of message M large, and simultaneously keeps the error, rate, i.e., the rate which messages are decoded incorrectly, small. The conditional distribution $P(Y|X)$ is a physical property of the channel itself, so it is not under the control of designer.

Fano's inequality has been widely used in lowering bounds of probability of transmission error through a communication channel. It is famous for linking the transmission error probability of a noisy communication channel to a standard information theoretic quantities including entropy and mutual information.

Let X and Y be the random variables representing the input and output, respectively, with the joint probability $P(x, y)$. Let e be the occurrence of error, i.e., that $X \neq \tilde{X}$, where $\tilde{X} = f(X)$ a noise approximate version of X. Fano's inequality is defined as

$$H(X|Y) \leq H(e) + P(e) \log(|\chi| - 1) \tag{2.61}$$

where χ is denoted as the support of X,

$$H(X|Y) = -\sum_{i,j} P(x_i, y_j) \log P(x_i|y_j) \tag{2.62}$$

is the conditional entropy,

$$P(e) = P(X \neq \tilde{X}) \tag{2.63}$$

is the probability of the communication error, and

$$H(e) = -P(e) \log P(e) - (1 - P(e)) \log(1 - P(e)) \tag{2.64}$$

is the corresponding binary entropy.

The inequality of Eq. (2.61) can be applied to lower bound the probability of support set recovery error. By making use of a bound $H(e) \leq 1$ and Fano's inequality, the decision error probability can be lower bounded as follows:

$$P(e) \geq \frac{H(X|Y) - 1}{\log|\chi| - 1} \tag{2.65}$$

Suppose $M \geq 2K$, then the map is one-to-one correspondent from X to Y. Therefore, $H(X|Y) = 0$ for $M \geq 2K$.

2.10 Sparse Convex Optimization

2.10.1 Introduction to Convex Optimization

If we want to know what is the convex *optimization problem*, first we need to know what is the *convex set*.

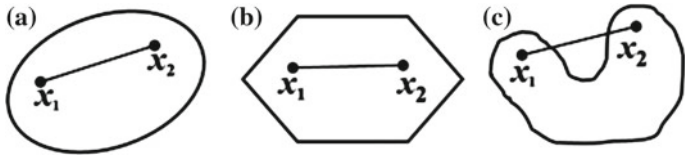

Fig. 2.7 Some examples about convex/nonconvex sets, where **a** and **b** are convex sets, and **c** is nonconvex set

Definition 2.10.1 (*Convex Set*) D is a set of $\mathbb{R}^N$. If any two points $\boldsymbol{x}_1, \boldsymbol{x}_2 \in D$ and $\lambda \in [0, 1]$ have the property as follows:

$$\lambda \boldsymbol{x}_1 + (1 - \lambda)\boldsymbol{x}_2 \in D.$$

We can call that this set D is a convex set, and $\lambda \boldsymbol{x}_1 + (1 - \lambda)\boldsymbol{x}_2$ is the convex combination of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$.

There are some examples, which are shown in Fig. 2.7 about the convex set in 2D space. We can note that if the set D is convex, and any two points $\boldsymbol{x}_1, \boldsymbol{x}_2 \in D$, we can get the segment which connect these two points also contained in set D.

After we have known what is the convex set, we need to know another important property of the convex optimization.

Definition 2.10.2 (*Convex Function*) The set $D \subset \mathbb{R}^N$ is a nonempty convex set, if $\forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in D$ and $\forall \alpha \in (0, 1)$ the following equation satisfied

$$f(\alpha \boldsymbol{x}_1 + (1 - \alpha)\boldsymbol{x}_2) \leq \alpha f(\boldsymbol{x}_1) + (1 - \alpha) f(\boldsymbol{x}_2),$$

we call the function $f(\boldsymbol{x})$ is the *convex function* in set D. If $' \leq'$ is replaced by $' <'$ in the above equation, we call the function $f(\boldsymbol{x})$ is *strictly convex function*.

From the definition of the convex function, we can see that the linear interpolation of two points on the convex function is not smaller than the function value. This can be seen from the Fig. 2.8 which is a scalar convex function.

There are some theorems about the convex function

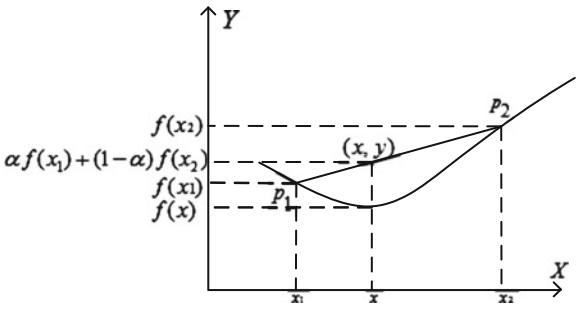

Fig. 2.8 The scaler convex function

Theorem 2.10.1 *If f_1, f_2 are convex functions in convex set D, and λ is a real number, we can get λf_1 and $f_1 + f_2$ are also convex functions.*

Theorem 2.10.2 *The set D is a convex set in $\mathbb{R}^N$, f is a convex function in set D, so f is continuous in set D.*

Theorem 2.10.3 *The set D is a nonempty convex set, f is a convex function in set D, we can get the following conclusions:*

(1) The set $D_\alpha = \{\boldsymbol{x} | \boldsymbol{x} \in D, f(\boldsymbol{x}) \leq \alpha\}$ is convex.
(2) The local minimal points of f in D is also the global minimal points of f, and the set of local minimal points is convex.

Theorem 2.10.4 *The set D is the nonempty open convex set, $f(\boldsymbol{x})$ is a differentiable function in set D. The $f(\boldsymbol{x})$ is the convex function if and only if $\forall \boldsymbol{x}, \boldsymbol{y} \in D$,*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + (\boldsymbol{y} - \boldsymbol{x})^T \nabla f(\boldsymbol{x}).$$

The theorems above tell us how to judge a function is convex function. The following definition defines the *convex optimization problem* by the convex function

Definition 2.10.3 (*Convex Optimization Problem*) For the optimization problem

$$\min f(\boldsymbol{x}) = f(x_1, x_2, \ldots, x_N), \ \boldsymbol{x} \in \mathbb{R}^N, \text{s.t. } g_j(\boldsymbol{x}) \leq 0, \quad j = 1, 2, \ldots, m \tag{2.66}$$

if the object function $f(\boldsymbol{x})$ and the constraint function $g_j(\boldsymbol{x})$ for $j = 1, 2, \ldots, m$ are convex function, we call this optimization problem is a convex optimization problem.

There are some properties about the convex optimization problem, before we present these properties, we need to know the definition of *feasible region*.

Definition 2.10.4 (*Feasible Region*) The feasible region of a optimization problem is the set of points that satisfy the constraint condition of this optimization problem.

Theorem 2.10.5 *The feasible region of the convex optimization problem $D = \{\boldsymbol{x} | g_j(\boldsymbol{x}) \leq 0, j = 1, 2, \ldots, m\}$ is convex.*

Theorem 2.10.6 *If give a point $\boldsymbol{x}_k$, then we can get the convex set*

$$S = \{\boldsymbol{x} | \boldsymbol{x} \in D, f(\boldsymbol{x}) \leq f(\boldsymbol{x}_k)\}$$

By the Theorem 2.10.6, we can imagine that if the object function is a function of two variables, the contour line of this function is the form of convex nested circles.

Theorem 2.10.7 *Any local optimal solution of the convex optimization problem is the global optimal solution.*

Theorem 2.10.8 *If the object function of the convex optimization problem $f(\boldsymbol{x})$ is the strictly convex function, the global optimal solution must be the only solution.*

2.10.2 Gradient, Subgradient, Accelerated Gradient

1. Gradient Descent

Gradient descent method is the first-order optimization algorithm. It is also called the *steepest descent*. The core of the gradient descent method is using the gradient descent direction iteratively at a fixed step to search for the local minimum of the object function.

The multivariable object function is differentiable in a neighborhood of a point $\boldsymbol{a}$, by the property of the gradient, and then the negative gradient direction $-\nabla F(\boldsymbol{a})$ is the decreasing fastest direction from $\boldsymbol{a}$. We can get that if

$$\boldsymbol{b} = \boldsymbol{a} - \gamma \nabla F(\boldsymbol{a}) \tag{2.67}$$

for γ small enough, then $F(\boldsymbol{a}) \geq F(\boldsymbol{b})$. Using this property, we start from a point $\boldsymbol{x}_0$ and have

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \gamma_n \nabla F(\boldsymbol{x}_n), \quad n \geq 0. \tag{2.68}$$

Further more, we can get

$$F(\boldsymbol{x}_0) \geq F(\boldsymbol{x}_1) \geq F(\boldsymbol{x}_2) \geq \ldots, \tag{2.69}$$

thus the sequence $\{\boldsymbol{x}_n\}$ can converge to the local minimum. The *step size* γ can be changed at every iteration. If F satisfy certain assumption, we can particularly choose the γ. This can guarantee to convergence to a local minimum, which is illustrated in Fig. 2.9.

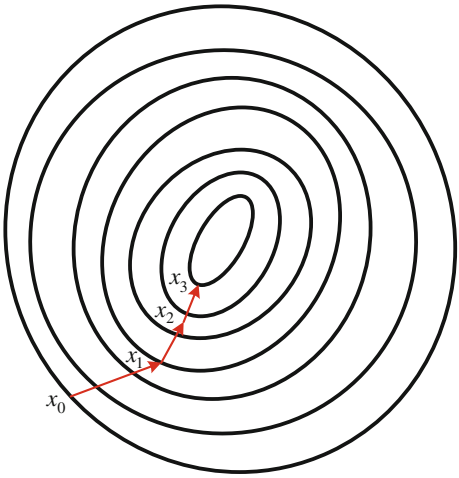

Fig. 2.9 Gradient descent with a constant step size γ

For example, the linear squares problem is used to minimize the following object function

$$F(\boldsymbol{x}_0) = \|A\boldsymbol{x} - \boldsymbol{y}\|_{\ell_2}^2. \tag{2.70}$$

This object function is smooth at every point, and we can compute its gradient as follows:

$$\nabla F(\boldsymbol{x}) = 2A^T(A\boldsymbol{x} - \boldsymbol{y}). \tag{2.71}$$

Using Eq. 2.71, we can iteratively find the points until the process convergence.

2. Subgradient Methods

In gradient descent or conjugate gradient methods, we need to compute the gradient of the object function. However, in case the object function is nonsmooth, we cannot use the gradient methods to solve optimization problem. We can see one model of the sparse representation problem,

$$\arg\min_{\boldsymbol{x}} = \|A\boldsymbol{x} - \boldsymbol{y}\|_{\ell_2}^2 + \lambda\|\boldsymbol{x}\|_{\ell_1}. \tag{2.72}$$

Because the ℓ_1-norm regularization term is nonsmooth, we cannot use the gradient methods. However, we can use the *subgradient methods*.

The *subgradient method* is a algorithm which can be used for minimization of the nondifferentiable convex function. There are some properties of subgradient method as follows [4]:

(1) The subgradient method applies directly to nondifferentiable F;
(2) The step lengths are not chosen by the line search as the ordinary gradient method. In the most common cases, the step lengths are fixed in advance;
(3) The subgradient method is not a descent method; the function value can increase.

The subgradient method is the first-order algorithm, and then it is much slower than the interior-point methods (or Newton's method in unconstrained case). Thus its performance depends on the scale and condition of the problem. However, subgradient method has its own advantages over the interior-point method and Newton methods.

(1) It can be applied to wider range of problems than interior-point methods and Newton methods;
(2) The memory requirement can be much smaller than interior-point methods and Newton methods, it can be used in extremely large scale problems;
(3) It can be combined with primal or dual decomposition technique to develop a distributed algorithm consequently.

Now, We introduce the concept of subgradient.

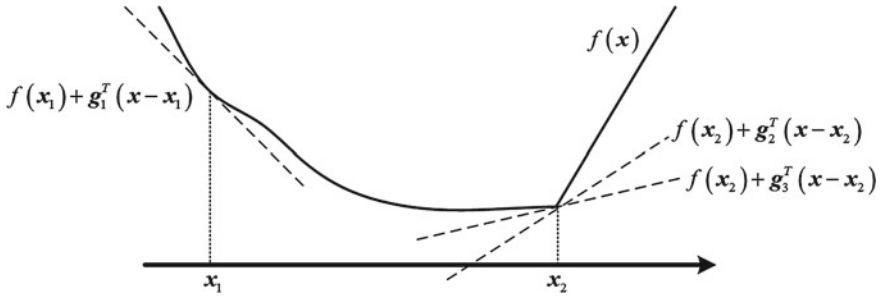

Fig. 2.10 The example of subgradient

Definition 2.10.5 (*Subgradient*) $\boldsymbol{g}$ is a subgradient of f (not necessarily convex) at point $\boldsymbol{x}$, if the following inequality is satisfied

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{g}^T(\boldsymbol{y} - \boldsymbol{x}), \forall \boldsymbol{y}. \tag{2.73}$$

One example of subgradient is illustrated in Fig. 2.10.

In the Fig. 2.10, $\boldsymbol{g}_2$, $\boldsymbol{g}_3$ are subgradients at $\boldsymbol{x}_2$, and $\boldsymbol{g}_1$ is a subgradient at $\boldsymbol{x}_1$. Note that some points have more than one subgradient. There are some further properties of subgradient:

(1) If f is convex, it has at least one subgradient at every point in domain of definition of f;
(2) If f is convex and differentiable, $\nabla f(\boldsymbol{x})$ is a subgradient of f at $\boldsymbol{x}$.

After giving the definition of *subgradient*, we shall give the definition of *subdifferential*.

Definition 2.10.6 (*Subdifferential*) The set of all subgradients of f at $\boldsymbol{x}$ is called the subdifferential of f at point $\boldsymbol{x}$, which can be noted as $\partial f(\boldsymbol{x})$.

From the definition of subdifferential, we can see that the subdifferential is a set. It also has some properties as follows:

(1) $\partial f(\boldsymbol{x})$ is a closed convex set;
(2) If f is convex and finite near the point $\boldsymbol{x}$, $\partial f(\boldsymbol{x})$ is nonempty;
(3) $\partial f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}$, if f is differentiable at $\boldsymbol{x}$;
(4) if $\partial f(\boldsymbol{x}) = \{\boldsymbol{g}\}$, then f is differentiable at point $\boldsymbol{x}$ and $\boldsymbol{g} = \nabla f(\boldsymbol{x})$.

We need to note that in many applications, we do not need to calculate $\partial f(\boldsymbol{x})$, but only need to find one $g \in \partial f(\boldsymbol{x})$. We can use concepts subgradient and subdifferential to solve the ℓ_1-norm optimization convex problem [21], which will be used in Sect. 5.2.1.

Now, we shall briefly review how to use gradient and subgradient method to solve the LASSO problem. By the subgradient method, we can easily get the unique solution of

$$\arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_{\ell_2}^2 + \lambda\|\boldsymbol{x}\|_{\ell_1} \tag{2.74}$$

for any $\boldsymbol{y} \in \mathbb{R}^D$, we have

$$x_i = S_\lambda(y_i), \tag{2.75}$$

where $S_\lambda(\cdot)$ is the *shrinkage operator* which is defined as

$$S_\lambda(y) = \operatorname{sgn}(y)\max\{|y|-\lambda, 0\}. \tag{2.76}$$

But for the LASSO problem Eq. (2.72), all elements of $\boldsymbol{x}$ are related by the matrix A. So, we cannot use the solution of Eq. (2.74) directly, but we can approximate the original object function by first-order Taylor expansion of $f(\boldsymbol{x}) = \|A\boldsymbol{x}-\boldsymbol{y}\|_{\ell_2}^2$ at the preceding point $\boldsymbol{x}_{k-1}$ and alternate the original LASSO problem as

$$\begin{aligned}\boldsymbol{x}_k = \arg\min_{\boldsymbol{x}}\Big\{ & f(\boldsymbol{x}_{k-1}) + \langle \boldsymbol{x}-\boldsymbol{x}_{k-1}, \nabla f(\boldsymbol{x}_{k-1})\rangle \\ & + \frac{1}{2t_k}\|\boldsymbol{x}-\boldsymbol{x}_{k-1}\|_{\ell_2}^2 + \lambda\|\boldsymbol{x}_1\|_{\ell_1}\Big\}.\end{aligned} \tag{2.77}$$

After ignoring the constant term, we can write the above equation as

$$\boldsymbol{x}_k = \arg\min_{\boldsymbol{x}}\left\{\frac{1}{2t_k}\|\boldsymbol{x}-(\boldsymbol{x}_{k-1}-t_k\nabla f(\boldsymbol{x}_{k-1}))\|_{\ell_2}^2 + \lambda\|\boldsymbol{x}_1\|_{\ell_1}\right\}. \tag{2.78}$$

Then, we can use the solution of Eq. (2.74) to get the solution as

$$\boldsymbol{x}_k = \arg\min_{\boldsymbol{x}} S_{\lambda t_k}(\boldsymbol{x}_{k-1}-t_k\nabla f(\boldsymbol{x}_{k-1})). \tag{2.79}$$

This is a fixed-point way to get the optimal solution, and it has two steps: (1) Using the gradient descent method to get the intermediate point; (2) Using the *shrinkage operator* to get $\boldsymbol{x}_k$. The process above is the *Iterative Shrinkage-Thresholding Algorithm* (ISTA) [12].

3. Accelerated Gradient

Even though the ISTA algorithm is very simple and adequate for the large scale problem, it converges very slowly. Toward this end, A. Beck et al. proposed a *Fast Iterative Shrinkage-Thresholding Algorithm* (FISTA) which has a higher convergence rate $O(1/k^2)$ by using the *Accelerate Gradient Descent* (AGD) method as compared with ISTA is $O(1/k)$.

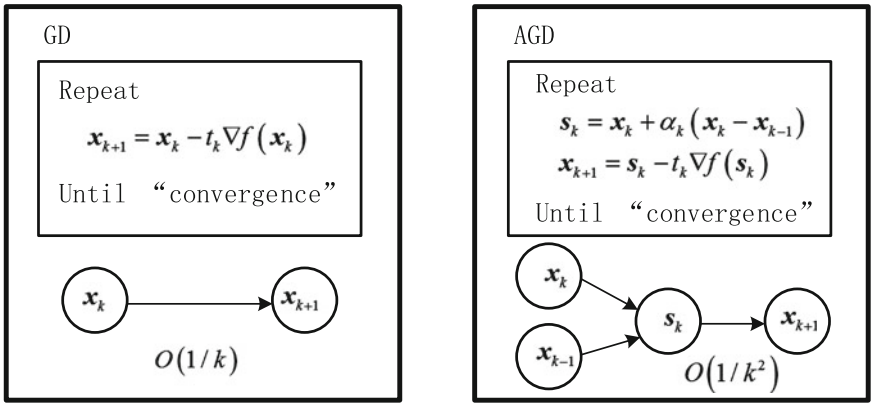

Fig. 2.11 The comparison between GD and AGD methods

The core idea of FISTA is using an accelerated gradient descent to replace the gradient descent step. The comparison of both gradient descent and accelerated gradient descent methods is illustrated in Fig. 2.11.

From Fig. 2.11, we can see that AGD method is using an intermediate variable $\boldsymbol{s}_k$ to get the final update $\boldsymbol{x}_{k+1}$. Using the AGD to replace the first step of ISTA, we can get the FISTA algorithm in sparse representation [3].

2.10.3 Augmented Lagrangian Method

Augmented Lagrangian methods are a certain class of algorithms to solve the constraint optimization problems. The core idea of the augmented Lagrangian methods is using the approximate unconstraint optimization problem to replace the constraint optimization problem. The difference between penalty methods and augmented Lagrangian methods is the augmented Lagrangian method that adds an additional term to the common penalty method's unconstraint object function. This difference can be seen from the following example.

We consider the optimization problem as follows:

$$\begin{aligned} &\arg\min_{\boldsymbol{x}} \quad f(\boldsymbol{x}) \\ &\text{s.t.} \quad g_i(\boldsymbol{x}) = 0, \quad i = 1, 2, \ldots, q. \end{aligned} \tag{2.80}$$

We only consider the equality constraints for simplicity. By the penalty method, we can approximate the original constraint problem by a unconstraint problem as follows:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_{i=1}^{q} \lambda_i g_i(\boldsymbol{x}). \tag{2.81}$$

We can also use the augmented Lagrangian method to approximate the original constraint problem as follows:

$$L_\rho(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_{i=1}^{q} \lambda_i g_i(\boldsymbol{x}) + \frac{\rho}{2} \sum_{i=1}^{q} g_i(\boldsymbol{x})^2. \tag{2.82}$$

From both approximate replacement by penalty method and augmented Lagrangian method of the original constraint optimization, we can find the augmented Lagrangian method adds an additional term $\frac{\rho}{2} \sum_{i=1}^{q} g_i(\boldsymbol{x})^2$, which is used for punishing the violations of the equality constraints $g_i(\boldsymbol{x})$. It has been proved that when ρ is large enough, the solution of unconstraint problem of augmented Lagrangian can coincides with the constrained solution of the original problem. The iterations of the algorithm end when the gradient $\rho g_i(\boldsymbol{x}) \Delta g_i(\boldsymbol{x}) = \mathbf{0}$. The whole algorithm of the augmented Lagrangian method is alternately updating $\boldsymbol{x}$ and $\boldsymbol{\lambda}$.

(1) Find the unconstraint minimum

$$\boldsymbol{x}^{(t+1)} = \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}, \boldsymbol{\lambda}). \tag{2.83}$$

(2) Update the multiplier vector $\boldsymbol{\lambda}$

$$\lambda_i^{(t+1)} = \lambda_i^{(t)} + \rho g_i(\boldsymbol{x}^{(t)}), \quad i = 1, \ldots, q \tag{2.84}$$

Using the augmented Lagrangian method, we solve the ℓ_1-norm convex optimization problem [20, 32]. Considering about the Basis Pursuit (BP) problem

Table 2.1 The Augmented Lagrangian algorithm flowchart

Input:	sensing matrix A, measurements $\boldsymbol{y}$, and parameter ρ
Initialization:	parameter vector $\boldsymbol{\lambda}$ with large value
while	(!stop criterion)
	Update the $\boldsymbol{x}$ as a LASSO problem
	$\boldsymbol{x}^{(t+1)} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_{\ell_1} + \langle \boldsymbol{\lambda}^{(t)}, A\boldsymbol{x} - \boldsymbol{y} \rangle + \frac{\rho}{2} \|A\boldsymbol{x} - \boldsymbol{y}\|_{\ell_2}^2$
	Update the parameter vector $\boldsymbol{\lambda}$
	$\boldsymbol{\lambda}^{(t+1)} = \boldsymbol{\lambda}^{(t)} + \rho(A\boldsymbol{x}^{(t+1)} - \boldsymbol{y})$
end while	
Output:	Coefficients $\boldsymbol{x}$

$$\begin{aligned} \arg\min_{\boldsymbol{x}} \quad & \|\boldsymbol{x}\|_{\ell_1}, \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{y} \end{aligned} \tag{2.85}$$

The algorithm flow chart of ALM is shown in Table 2.1.

References

1. Achlioptas, D.: Database-friendly random projections: Johnson-Lindenstrauss with binary coins. J. Comput. Syst. Sci. **66**(4), 671–687 (2003)
2. Baraniuk, R., Davenport, M., DeVore, R., Wakin, M.: A simple proof of the restricted isometry property for random matrices. Constr. Approx. **28**(3), 253–263 (2008)
3. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**(1), 183–202 (2009)
4. Boyd, S., Xiao, L., Mutapcic, A.: Subgradient Methods. Lecture, Stanford University, Autumn Quarter **54**(1), 48–61 (2003)
5. Candes, E.J.: The restricted isometry property and its implications for compressed sensing. Comptes Rendus Mathematique **346**(9), 589–592 (2008)
6. Candes, E.J., Romberg, J.K., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. Commun. Pure Appl. Math. **59**(8), 1207–1223 (2006)
7. Candes, E.J., Tao, T.: Decoding by linear programming. IEEE Trans. Inf. Theory **51**(12), 4203–4215 (2005)
8. Candes, E.J., Tao, T.: Near-optimal signal recovery from random projections: universal encoding strategies? IEEE Trans. Inf. Theory **52**(12), 5406–5425 (2006)
9. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM J. Sci. Comput. **20**(1), 33–61 (1998)
10. Cohen, A., Dahmen, W., DeVore, R.: Instance optimal decoding by thresholding in compressed sensing. Technical Report DTIC Document (2008)
11. Cohen, A., Dahmen, W., DeVore, R.: Compressed sensing and best k-term approximation. J. Am. Math. Soc. **22**(1), 211–231 (2009)
12. Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Commun. Pure Appl. Math. **57**(11), 1413–1457 (2004)
13. Davenport, M.A.: Random observations on random observations: Sparse signal acquisition and processing. Ph.D. thesis. Citeseer (2010)
14. Davenport, M.A., Duarte, M.F., Eldar, Y.C., Kutyniok, G.: Introduction to compressed sensing 93 (2011)
15. Donoho, D.L., Elad, M.: Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1-minimization. Proc. Natl. Acad. Sci. **100**(5), 2197–2202 (2003)
16. Donoho, D.L., Huo, X.: Uncertainty principles and ideal atomic decomposition. IEEE Trans. Inf. Theory **47**(7), 2845–2862 (2001)
17. Donoho, D.L., Stark, P.B.: Uncertainty principles and signal recovery. SIAM J. Appl. Math. **49**(3), 906–931 (1989)
18. Elad, M.: Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing. Springer, New York (2010)
19. Eldar, Y.C., Kutyniok, G.: Compressed Sensing: Theory and Applications. Cambridge University Press, Cambridge (2012)
20. Goldstein, T., Osher, S.: The split Bregman method for ℓ_1-regularized problems. SIAM J. Imaging Sci. **2**(2), 323–343 (2009)
21. Hale, E.T., Yin, W., Zhang, Y.: Fixed-point continuation for ℓ_1-minimization: methodology and convergence. SIAM J. Optim. **19**(3), 1107–1130 (2008)

22. Haupt, J., Nowak, R.: Signal reconstruction from noisy random projections. IEEE Trans. Inf. Theory **52**(9), 4036–4048 (2006)
23. Johnson, W.B., Lindenstrauss, J.: Extensions of Lipschitz mappings into a Hilbert space. Contemp. Math. **26**(189–206), 1 (1984)
24. Lee, H.N.: Introduction to Compressed Sensing. Lecture Notes. Springer (2011)
25. Li, P., Hastie, T.J., Church, K.W.: Very sparse random projections. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2006)
26. Muthukrishnan, S.: Data Streams: Algorithms and Applications. Now Publishers Inc. (2005)
27. Needell, D., Tropp, J.A.: CoSaMP: iterative signal recovery from incomplete and inaccurate samples. Commun. ACM **53**(12), 93–100 (2010)
28. Shannon, C.E.: Communication in the presence of noise. IEEE Proc. IRE **37**(1), 10–21 (1949)
29. Sharon, Y., Wright, J., Ma, Y.: Computation and relaxation of conditions for equivalence between ℓ_1 and ℓ_0 minimization. IEEE Trans. Inf. Theory **5** (2007)
30. Wang, H., Nie, F., Huang, H.: Multi-view clustering and feature learning via structured sparsity. In: ICML (2013)
31. Yanning Shen, J.F., Li, H.: Exact reconstruction analysis of log-sum minimization for compressed sensing. IEEE Signal Process. Lett. **20**(12), 1223–1226 (2013)
32. Yin, W., Osher, S., Goldfarb, D., Darbon, J.: Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing. SIAM J. Imaging Sci. **1**(1), 143–168 (2008)
33. Zhang, K., Zhang, L., Yang, M.H.: Real-time compressive tracking. In: ECCV. Springer (2012)
34. Zhou Zhou, K.L., Fang, J.: Bayesian compressive sensing using normal product priors. IEEE Signal Process. Lett. **22**(5), 583–587 (2015)

http://www.springer.com/978-1-4471-6713-6

Sparse Representation, Modeling and Learning in Visual Recognition
Theory, Algorithms and Applications
Cheng, H.
2015, XIV, 257 p. 73 illus., Hardcover
ISBN: 978-1-4471-6713-6