Chapter 2
How (Well) Compressed Sensing Works in Practice

2.1 Non-Worst-Case Assessment of CS Performance

One of the main problems with coherence and restricted isometries is that the corresponding parameters are explicitly calibrated on worst-case scenarios. This corresponds to the desire of providing guarantees, thanks to which the system is known to operate correctly. Yet, acquisition systems work on random inputs and it is perfectly sensible to characterize their performance by probabilistic means. This is particularly true when the input is not their unique random components since, for example, the matrix $\boldsymbol{A}$ is also a possibly time varying, uncertain ingredient of the processing.

An instructive example of this alternative route is given by a more geometric approach to the properties of the minimization problem (1.27) (Basis Pursuit—BP) and to its relationship with the minimization problem (1.26). Everything hinges on a special kind of polytopes.

Definition 2.1 These are the definitions we need to proceed in our analysis

- A p-dimensional convex polytope $P \subset \mathbb{R}^p$ is the convex hull of a set V of points in $\mathbb{R}^p$.
- If no point in V can be dropped without changing the resulting convex hull, then the points in V are the *vertices* of P.
- The intersection $P \cap h$ of a p-dimensional convex polytope P with a $p-1$-dimensional hyperplane that does not contain any point of the interior of P is called a *facet* of P. Facets can be 0-dimensional (vertices), 1-dimensional (edges), or in general q-dimensional with $q < p$.
- Given a convex polytope $P \subset \mathbb{R}^p$ with vertices $\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots$ and a $q \times p$ matrix $\boldsymbol{B}$, the convex hull of the points $\boldsymbol{B}\boldsymbol{v}_0, \boldsymbol{B}\boldsymbol{v}_1, \ldots$ is a q-dimensional convex polytope $Q = \boldsymbol{B}P \subset \mathbb{R}^q$.

M. Mangia et al., *Adapted Compressed Sensing for Effective Hardware Implementations*, DOI 10.1007/978-3-319-61373-4_2

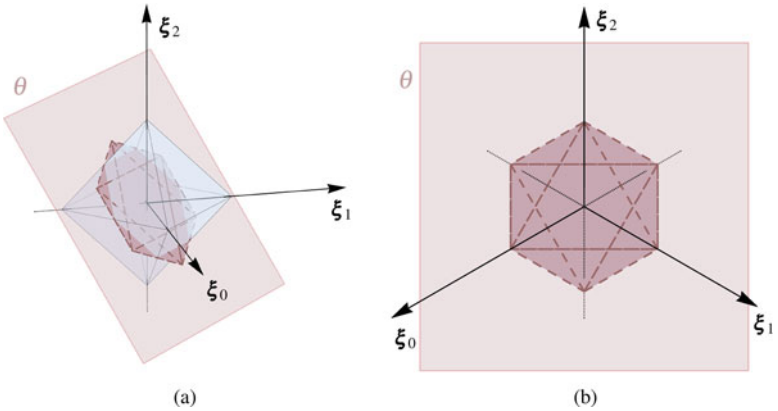

Fig. 2.1 Construction of $\boldsymbol{B}S_3^1(1)$ starting from $S_3^1(1)$ using $\boldsymbol{B}$ as in (1.7) (**a**) and its frontal view allowing face counting (**b**)

- A polytope is said to be *centrosymmetric* if $\boldsymbol{v} \in V$ implies $-\boldsymbol{v} \in V$.
- If $\boldsymbol{e}_j = (0, \ldots, 0, 1, 0, \ldots, 0)^\top$ where the unique 1 appears in the j-th position, then the *crosspolytope* in $\mathbb{R}^p$ is defined as the centrosymmetric convex hull of $V = \{\pm\boldsymbol{e}_0, \pm\boldsymbol{e}_1, \ldots, \pm\boldsymbol{e}_{p-1}\}$. In our previous notation the crosspolytope is nothing but $S_p^1(1)$.

Starting from these definitions that are standard elements in the theory of convex polytopes, one may develop specific concepts related to possibility of reconstructing the original signal from the measurements vector [7, 8, Theorem 7.5].

Definition 2.2 A p-dimensional centrosymmetric polytope is said to be *centrally q-neighborly* if every subset of V with q elements that does not include two antipodal vertices is the set of vertices of a $q-1$-dimensional facet of P.

Theorem 2.1 *If $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$ has a unique solution with not more than κ non-null components, then such a solution is the unique solution of BP if and only if $\boldsymbol{B}S_d^1(1)$ has $2d$ vertices and is centrally $(\kappa-1)$-neighborly.*

The point of interest in Theorem 2.1 is that it gives a necessary and sufficient condition for signal reconstruction: no worst-case bounding is involved. This has a substantial impact on the predictability of CS performance. As an example, we may go back to the matrix $\boldsymbol{B}$ in (1.7) and recall that its mutual coherence equal to 1 and its RIC equal to $1/2$ prevent the application of the results leveraging those concepts, i.e., of Theorems 1.5, 1.6, and 1.7.

Yet, Theorem 2.1 explains why reconstruction by means of BP is always effective in the noiseless case. Figure 2.1a reports the construction of $\boldsymbol{B}S_3^1(1)$ starting from $S_3^1(1)$. The same $\boldsymbol{B}S_3^1(1)$ is visualized in Fig. 2.1b. In this case $d = 3$ and $\kappa = 1$ and it is easy to verify that $\boldsymbol{B}S_3^1(1)$ has $2d = 6$ vertices each of them trivially being a face so that the polytope is centrally 0-neighborly.

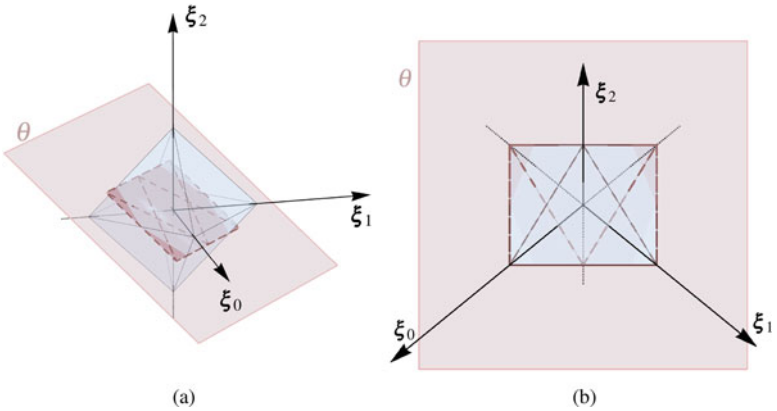

Fig. 2.2 Construction of $\boldsymbol{B}S_3^1(1)$ starting from $S_3^1(1)$ using $\boldsymbol{B}$ as in (2.1) (**a**) and its frontal view allowing face counting (**b**)

The same theorem indicates when the choice of $\boldsymbol{B}$ may prevent us from getting a reconstruction. As an example consider

$$\boldsymbol{B} = \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{pmatrix} \tag{2.1}$$

and the resulting $\boldsymbol{B}S_3^1(1)$ as in Fig. 2.2. In this case, the number of vertices is only $4 < 2d = 6$ and Theorem 2.1 implies that reconstruction by means of (1.27) may be impossible since the sparsity prior may not be enough to select a unique solution of $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$. What happens is described in Fig. 2.2 in which $S_3^1(1)$ is projected with the new $\boldsymbol{B}$ on a new plane θ. Two out of 8 faces of $S_3^1(1)$ are orthogonal to θ so that one of the vertices of each of these two faces gets mapped into a point on the projection of the edge connecting the other two vertices, disappearing in the final polytope $\boldsymbol{B}S_3^1(1)$.

This is what prevents reconstruction. In fact, assume that you want to recover the same point $\boldsymbol{\xi}'''$ as in Fig. 1.14a. The straight line corresponding to $\boldsymbol{y}''' = \boldsymbol{B}\boldsymbol{\xi}$ that is orthogonal to θ is also parallel to 2 faces of $S_3^1(1)$ so that its intersection with $S_3^1(\|\boldsymbol{\xi}'''\|_1)$ is a whole segment, each point of which is a solution of BP. This is visualized in Fig. 2.3.

As a further application to our toy case, the very same Theorem 2.1 explains why, no matter how the plane θ is positioned, the solution of BP is not able to retrieve $\boldsymbol{\xi}$ from $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$ when it is known that $\boldsymbol{\xi}$ is 2-sparse instead of 1-sparse. In fact, to have $2d$ vertices, $\boldsymbol{B}S_3^1(1)$ must be a hexagon. Yet, only the pairs of consecutive vertices belong to a $(\kappa - 1)$-dimensional facet (that for $\kappa = 2$ is an edge) of a hexagon, while other pairs do not, preventing central neighborliness. Hence, no matter how θ is positioned, BP cannot be used for signal reconstruction.

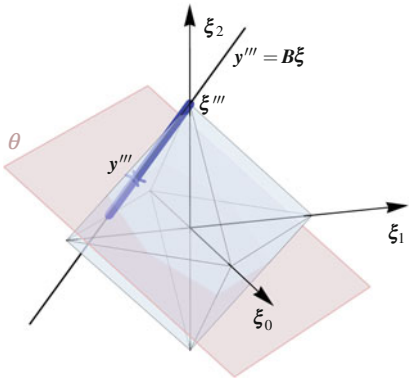

Fig. 2.3 Trying the reconstruction of $\boldsymbol{\xi}'''$ starting from $\boldsymbol{y}''' = \boldsymbol{B}\boldsymbol{\xi}'''$ by means of BP. All the points in the *blue thick segment* are equally good solutions

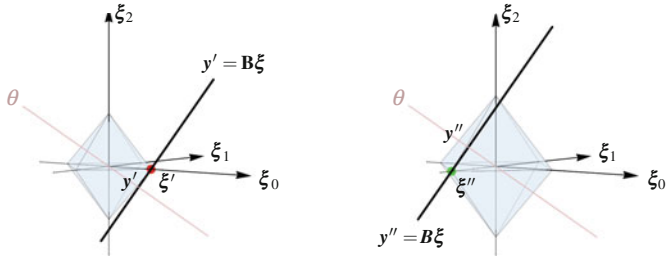

Fig. 2.4 Successful reconstruction of $\boldsymbol{\xi}'$ starting from $\boldsymbol{y}' = \boldsymbol{B}\boldsymbol{\xi}'$ (**a**) and of $\boldsymbol{\xi}''$ from $\boldsymbol{y}'' = \boldsymbol{B}\boldsymbol{\xi}''$ (**b**) by means of BP

Lastly, this powerful point of view can be extended to the case in which the signal to retrieve is random. In fact, Theorem 2.1 is a guarantee independent of $\boldsymbol{\xi}$, that ceases to hold if even a single $\boldsymbol{\xi}$ cannot be reconstructed. Yet, even when the guarantee does not hold, like for the matrix in (2.1), there are signals that can be reconstructed.

In particular, in passing from $S_3^1(1)$ to $\boldsymbol{B}S_3^1(1)$, 2 out of 6 vertices are lost and the signals $\boldsymbol{\xi}$ that cannot be retrieved are exactly those at the corresponding vertices of $S_3^1(\|\boldsymbol{\xi}\|_1)$, like $\boldsymbol{\xi}'''$ in Fig. 2.3. Yet, Fig. 2.2 shows that two other pairs of vertices appear in $\boldsymbol{B}S_3^1(1)$ and signals on the corresponding vertices of $S_3^1(\|\boldsymbol{\xi}\|_1)$ can still be reconstructed. This is shown in Figs. 2.4a and b where the same $\boldsymbol{\xi}'$ and $\boldsymbol{\xi}''$ as in Fig. 1.6 are uniquely identified by the intersection of the straight line $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$ and the minimum radius $\|\cdot\|_1$ ball.

Intuitively speaking, if the original 1-sparse signal $\boldsymbol{\xi}$ has the same probability of aligning with each of the axes, the probability that BP is effective in recovering it is equal to the ratio of the number of surviving vertices over the number of original vertices, i.e., $4/6 = 2/3$.

All this can be generalized to cope with a larger sparsity κ. To understand how, we may first define $\phi_k(\cdot)$ as the operator that counts the number of κ-dimensional facets of its polytope argument and state the following [7, Theorem 3].

Theorem 2.2 *Let $\boldsymbol{B}$ be an $m \times d$ matrix such that if $\boldsymbol{B}\boldsymbol{\alpha} = 0$ for a vector $\boldsymbol{\alpha}$ with less than m nonzeros then $\boldsymbol{\alpha} = 0$. Let also $\kappa < m/2$.*

Given a subset $K \subset \{0, \ldots, d-1\}$ of cardinality κ, we may have that if $\text{supp}(\boldsymbol{\xi}) = K$ then $\boldsymbol{\xi}$ can be reconstructed from $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$ by means of BP. Indicate with K_{BP} the number of such subsets, and with $K_{\text{tot}} = \binom{d}{\kappa}$ the total number of possible subsets of cardinality κ. Then

$$\frac{K_{\text{BP}}}{K_{\text{tot}}} \geq \frac{\phi_{\kappa-1}(\boldsymbol{B}S_d^1(1))}{\phi_{\kappa-1}(S_d^1(1))} \tag{2.2}$$

Assuming that the original signal has the same probability of featuring any of the K_{tot} supports of cardinality κ, the above result can be immediately recast into probabilistic terms to say that $p_{\text{BP}} = K_{\text{BP}}/K_{\text{tot}}$ is the probability of successful reconstruction by means of BP and is not less than the ratio of facets counts in (2.2). A dual result is available for the case in which $\boldsymbol{B}$ is random [8, Theorem 7.7].

Theorem 2.3 *Let $\boldsymbol{B}$ be a random $m \times d$ matrix whose probability distribution is invariant for any signed permutation of rows. Let $\boldsymbol{\xi} \in \mathbb{R}^d$ be a κ-sparse vector and $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$ the corresponding random measurement vector. The probability p_{BP} that BP retrieves $\boldsymbol{\xi}$ from $\boldsymbol{y}$ is bounded by*

$$p_{\text{BP}} \geq \frac{\mathbf{E}[\phi_{\kappa-1}(\boldsymbol{B}S_d^1(1))]}{\phi_{\kappa-1}(S_d^1(1))}$$

Note that, in general, facets counting is a combinatorial task so that the computation of $\phi_{\kappa-1}(\cdot)$ in high-dimensional settings can be expensive if not impossible. From this point of view, the introduction of random matrices $\boldsymbol{B}$ can be helpful if paired with the asymptotic conditions that are the mathematical equivalent of the high-dimensional setting in which CS is applied. Many sophisticated results are born in this area, whose simplest prototype is probably the one that we rephrase here in our terms [10].

Theorem 2.4 *Let $\boldsymbol{B} \sim \text{RGE}\,(\text{iid})$ with unit variance entries, $d = (\texttt{DR} \times \texttt{CR})m$, and $m = \texttt{OH}\,\kappa$. There is a function $\psi(\cdot)$ such that*

$$\lim_{d\to\infty} \frac{\phi_{\kappa-1}(\boldsymbol{B}S_d^1(1))}{\phi_{\kappa-1}(S_d^1(1))} = \begin{cases} 1 & \text{if } \texttt{DR} \times \texttt{CR} < \psi(\texttt{OH}) \\ 0 & \text{if } \texttt{DR} \times \texttt{CR} > \psi(\texttt{OH}) \end{cases}$$

Collecting the results in Theorems 2.2, 2.3, and 2.4 one gets that, as the dimensionality increases, there is a crisp *phase transition* in the possibility of

reconstructing $\boldsymbol{\xi}$ from its random projections. The excess of measurements with respect to the actual degrees of freedom in the signal (OH) controls the possibility of accommodating a certain dimensionality reduction DR $\times$ CR while maintaining the retrievability of the original signal.

Though Theorem 2.4 leverages RGE (iid), the existence and shape of the function ψ has been empirically found to be a general property [9] when the entries of $\boldsymbol{B}$ are iid or its rows are an iid random subset of certain orthonormal basis.

In the noiseless and exactly sparse case, this makes polytope-based analysis much closer to real performance than coherence or RIP-based considerations since neither the finite-dimension results, nor asymptotics of random $\boldsymbol{B}$ rely on worst-case bounding.

As a consequence, though neither the theory which heavily relies on symmetry considerations, nor the empirical evidence gathered so far in the Literature, say much on the possibility of straightforwardly applying this point of view to the design of a proper sensing matrix $\boldsymbol{A}$, we know that if n is large and we increase m enough, CS will eventually work very well (the *probability 1* implicit in Theorem 2.4).

From a more engineering point of view, this can be reversed to say that our aim is to find the minimum possible m for which CS works very well. Properly evolved and specialized, this is the key idea behind the discussion in the chapters to follow.

2.2 Beyond Basis Pursuit

Despite its theoretical appeal, BP is only an archetypal reconstruction method. In practical terms BP and its denoising variant BPDN have been implemented with a variety of methods, ranging from straightforward mapping to classical mathematical programming problems leveraging linear and quadratic optimization tools, to specialized procedures that look at them as a particular case of a convex optimization task.

The activity in this field revealed, for example, that it is sometimes convenient to address the BP or BPDN problems not in the *synthesis form* contained in (1.27) but in an alternative *analysis form*.

Note, in fact, that (1.27) depends only on $\boldsymbol{B}$ and thus considers and seeks to reconstruct the signal $\boldsymbol{\xi}$ in the sparsity domain. Once that $\boldsymbol{\xi}$ is known one may *synthesize* $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$.

Assume now that a linear operator $\boldsymbol{D}^*$ is available such that if $\boldsymbol{\xi} = \boldsymbol{D}^*\boldsymbol{x}$ then $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$. When $\boldsymbol{D}$ is a non-singular square matrix we simply have $\boldsymbol{D}^* = \boldsymbol{D}^{-1}$ and when $\boldsymbol{D}$ is a frame, $\boldsymbol{D}^*$ is the dual frame operator. With this, we may concentrate directly on the true signal $\boldsymbol{x}$ and try to solve the "equivalent"

$$\begin{aligned} &\arg\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{D}^*\boldsymbol{x}\|_1 \\ &\quad \text{s.t. } \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2 \leq \epsilon \end{aligned} \tag{2.3}$$

Clearly, (1.27) and (2.3) are not equivalent when $\boldsymbol{D}$ is not an invertible matrix. In fact, $\boldsymbol{D}^*\boldsymbol{\xi}$ is only one of the many possible representations of $\boldsymbol{x}$ in the sparsity domain and is the only one considered while scanning the feasibility space of (2.3) while (1.27) considers all of them. What happens is that the choice made by $\boldsymbol{D}^*$ acts as a further prior and in this role, it is often useful to decreased dimensionality of the analysis form of BP and BPDN and help them finding good solutions.

Beyond this, accounting for all the methods and implementations described in the Literature and/or made available to practitioners is out of the scope of this book. Yet, it is useful to mention some of the most widespread tools distinguishing between those helping the implementation of BP, BPDN, and their other variants, those tackling the reconstruction problem from a theoretically different point of view, and those that are mainly based on heuristic considerations and yield lightweight iterative procedures that may be extremely useful when the resources dedicated to signal retrieval are limited.

Further to their implementation in commercial, large-scale solvers, BP and BPDN can be solved by quite a few implementations. Among them it is worthwhile mentioning those in Table 2.1 where we give the commonly used acronym, a pointer to some ready-to-use code and references to the relevant Literature.

Since BP and BPDN are convex optimization problems, they can be tackled by convex solvers with wider applicability. Those in Table 2.2 are particularly effective in modeling and solving the two standard reconstruction problems. Additionally, their greater generality can be used to add constraints that model priors further to sparsity that may available on the signal, thus increasing reconstruction performance.

Further to these methods, instead of depending on the $\|\cdot\|_1$ norm and its favorable geometry, signal reconstruction can be approached from completely different points of view, e.g., from the estimation, or machine learning, or regression point of view. Different approaches result in different algorithms some of which are listed in Table 2.3.

Table 2.1 Some dedicated BP/BPDN solvers

Solver	Url	Reference
SPGL1	www.math.ucdavis.edu/~mpf/spgl1/	[2]
NESTA	statweb.stanford.edu/~candes/nesta/	[1]

Table 2.2 Some solvers of convex optimization problems that can be used for signal retrieval

Solver	Url	Reference
CVX	cvxr.com/	[12, 13]
Unlocbox	lts2.epfl.ch/unlocbox/	[5]

Table 2.3 Some signal reconstruction methods based on various heuristic

Solver	Url	Reference
GAMP	gampmatlab.wikia.com	[17]
IRLS	http://stemblab.github.io/irls/	[6]
SBL	dsp.ucsd.edu/~zhilin/BSBL.html	[14]

Table 2.4 Some signal reconstruction methods based on various heuristic

Solver	Url	Reference
FOCUSS	dsp.ucsd.edu/~jfmurray/software.htm	[11]
OMP	http://www.mathworks.com/matlabcentral/fileexchange/32402-cosamp-and-omp-for-sparse-recovery	[16]
CoSaMP		[16]
Iterative hard thresholding	www.personal.soton.ac.uk/tb1m08/sparsify/sparsify.html	[3]
	http://sparselab.stanford.edu/	

Table 2.5 Code sketch for CoSaMP

Require: $\boldsymbol{y}$ vector of measurements
Require: κ sparsity level
Require: $\boldsymbol{B} = \boldsymbol{AD}$ sensing matrix
  $\hat{\boldsymbol{\xi}} \leftarrow 0$ ▷ signal guess
  $\Delta\boldsymbol{y} \leftarrow \boldsymbol{y} - \boldsymbol{B}\hat{\boldsymbol{\xi}} = \boldsymbol{y}$ ▷ error in reproducing measurements
  repeat
    $\Delta\boldsymbol{\xi} \leftarrow \boldsymbol{B}^\top \Delta\boldsymbol{y}$ ▷ error in signal guess
    $J = \text{supp}(\hat{\boldsymbol{\xi}}) \cup \text{supp}(\Delta\boldsymbol{\xi}^{2\kappa\uparrow})$ ▷ support to correct error in signal guess
    $\hat{\boldsymbol{\xi}} \leftarrow 0$
    $\hat{\boldsymbol{\xi}}_J = \left(\boldsymbol{B}_{\cdot,J}\right)^\dagger \boldsymbol{y}$
    $\hat{\boldsymbol{\xi}} \leftarrow \hat{\boldsymbol{\xi}}^{\kappa\uparrow}$ ▷ new signal guess
    $\Delta\boldsymbol{y} = \boldsymbol{y} - \boldsymbol{B}\hat{\boldsymbol{\xi}}$ ▷ new error in reproducing measurements
  until convergence

Finally, procedures exist that retrieve the original signal by considering that the main issue in the computation of $\boldsymbol{\xi}$ is not finding a generic solution to $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{\xi}$ but to find the sparse one. Starting from this, it is possible to generate solutions iteratively adjusting their sparsity at each step. Different heuristics may be used to promote sparsity and give raise to different methods, some of which are listed in Table 2.4. The simple structure of these methods and their relatively good performance make them ideal for CS embodiments in which the resources devoted to signal reconstruction are limited.

As an example of how simple such algorithms can be, assume that $\boldsymbol{B}$ is well approximated by a random matrix with i.i.d., zero-average, entries and that the $\|\cdot\|_2$ norm of each column is approximately equal. Since the columns of $\boldsymbol{B}$ are independent, the matrix $\boldsymbol{B}^\top\boldsymbol{B}$ is well approximated by a diagonal matrix.

Assume now that an estimate $\hat{\boldsymbol{\xi}}$ is given of the true $\boldsymbol{\xi}$. The measurement vector corresponding to $\hat{\boldsymbol{\xi}}$ is $\hat{\boldsymbol{y}} = \boldsymbol{B}\hat{\boldsymbol{\xi}}$ whose difference with respect to the true measurement vector is $\Delta\boldsymbol{y} = \boldsymbol{B}(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi})$. Thanks to the previous considerations on $\boldsymbol{B}^\top\boldsymbol{B}$, we also have that $\boldsymbol{B}^\top\Delta\boldsymbol{y} = \boldsymbol{B}^\top\boldsymbol{B}(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}) \approx \|\boldsymbol{B}_{\cdot,0}\|_2\,(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi})$.

Hence, the largest nonzero components of $\boldsymbol{B}^\top\Delta\boldsymbol{y}$ indicate the components of the signal that have been mistaken most by taking $\hat{\boldsymbol{\xi}}$ instead of $\boldsymbol{\xi}$. This is the core step in the CoSaMP algorithm whose complete definition is given in Table 2.5 where: $\cdot^{p\uparrow}$ that takes a vector and gives its thresholded version in which all but the p

largest component are set to zero, $\cdot^{\dagger}$ indicates the Moore–Penrose pseudo-inverse of a matrix, and given an index set J, a vector $\boldsymbol{v}$, and a matrix $\boldsymbol{M}$, $\boldsymbol{v}_J$ is the subvector of $\boldsymbol{v}$ containing only the entries of $\boldsymbol{v}$ with indexes in J, while $\boldsymbol{M}_{\cdot J}$ is the submatrix of $\boldsymbol{M}$ made of the columns of $\boldsymbol{M}$ whose indexes stay in J.

Though the convergence criterion is not specified, it is clear that the procedure itself is much simpler than solving a convex optimization problem. This is the reason why methods like this and like the others in Table 2.4 are often used in limited-resources realizations of reconstruction stages (see, e.g., [4]).

2.3 A Framework for Performance Evaluation

In the light of the discussions in Chap. 1 and of the initial section of this chapter, it is easy to state that a precise assessment of the performance of a CS system is far from easy.

An obvious intuition is that performance must be related to the magnitude of the reconstruction error, i.e., to the difference between the true sparse representation ξ and the one estimated by the reconstruction algorithm $\hat{\boldsymbol{\xi}}$ or between the true signal $\boldsymbol{x}$ and $\hat{\boldsymbol{x}} = \boldsymbol{D}\hat{\boldsymbol{\xi}}$.

Yet, the classical theory of Chap. 1 follows a worst-case leitmotif and gives bounds on quantities like $\left\| \hat{\boldsymbol{\xi}} - \boldsymbol{\xi} \right\|_2$ that are either rarely applicable (for example, because they pose too strict requirements on measurement matrices $\boldsymbol{A}$) or quite loose and ultimately very far from actual behavior.

Even the non-worst-case approach described at the beginning of this chapter has problems since its face-counting argument, though allowing a much sharper distinction between what can be reconstruct and what cannot, scales poorly as dimension increase and cannot be applied in practice.

Last but not least, the construction of the matrices $\boldsymbol{A}$ is often done by random means. This, paired with the intrinsic random nature of the signal to acquire, implies that reconstruction error is a quite complicated random quantity.

The most straightforward way of addressing all these problems is to resort to extensive Montecarlo simulations. Such an approach is the most common both in the Literature and in practice and consists in generating a large number W of signal instances $\boldsymbol{x}^{(j)}$ and of measurement matrices $\boldsymbol{A}^{(j)}$ for $j = 0, \ldots, W-1$, use each of them to compute $\boldsymbol{y}^{(j)} = \boldsymbol{A}^{(j)}\boldsymbol{x}^{(j)}$ and then run one of the algorithms mentioned before to compute the estimation $\hat{\boldsymbol{x}}^{(j)}$ and consequently the reconstruction error in that case. The statistic of such an error is usually summarized in single numbers by means of one of the two approaches.

To begin with, it is most natural to define a Reconstruction Signal-to-Noise-Ratio

$$\texttt{RSNR[dB]} = 20 \log_{10} \left(\frac{\|\boldsymbol{x}\|_2}{\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2} \right)$$

that acts as a merit figure, i.e., the larger the RSNR[dB], the better the reconstruction.

Then one may try to estimate the Average RSNR[dB] as

$$\texttt{ARSNR[dB]} = \mathbf{E}\left[20\log_{10}\left(\frac{\|\boldsymbol{x}\|_2}{\|\hat{\boldsymbol{x}}-\boldsymbol{x}\|_2}\right)\right] \approx \frac{1}{W}\sum_{j=0}^{W-1} 20\log_{10}\left(\frac{\left\|\boldsymbol{x}^{(j)}\right\|_2}{\left\|\hat{\boldsymbol{x}}^{(j)}-\boldsymbol{x}^{(j)}\right\|_2}\right) \tag{2.4}$$

Alternatively, one may assume that the reconstruction is correct when the corresponding RSNR[dB] exceeds a certain $\texttt{RSNR[dB]}_{\min}$ and define a Probability of Correct Reconstruction (PCR) as

$$\texttt{PCR} = \Pr\{\texttt{RSNR[dB]} \geq \texttt{RSNR[dB]}_{\min}\} \approx \frac{\left|\left\{\frac{\left\|\boldsymbol{x}^{(j)}\right\|_2}{\left\|\hat{\boldsymbol{x}}^{(j)}-\boldsymbol{x}^{(j)}\right\|_2} \geq 10^{\frac{\texttt{RSNR[dB]}_{\min}}{20}}\right\}\right|}{W}$$

Clearly, ARSNR[dB] and PCR are general-purpose merit figures and real-world applications may provide more significant indexes for establishing the acquisition performance. When applications are addressed at the end of this book, those merit figures will be possibly described and applied.

Yet, examples made to describe the adaptive method we address will use ARSNR[dB] and PCR and a uniform framework for the accumulation of Montecarlo trials.

In particular, we are interested in n-dimensional signals $\boldsymbol{x}$ that are both localized and κ-sparse with respect to a certain reference system $\boldsymbol{D}$ that we assume to be an orthonormal basis.

To generate samples of $\boldsymbol{x}$ we start from an instance of a zero-mean Gaussian random vector $\boldsymbol{x}'$ with covariance/correlation matrix $\mathscr{X}'$ and do the following steps:

$\boldsymbol{x}' \sim \mathrm{N}\left(0, \mathscr{X}'\right)$
$\boldsymbol{\xi}' \leftarrow \boldsymbol{D}^{-1}\boldsymbol{x}' = \boldsymbol{D}^{\top}\boldsymbol{x}'$
$\boldsymbol{\xi} \leftarrow (\boldsymbol{\xi}')^{\kappa\uparrow}$
$\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$

that formalize the intuitive idea of taking a possibly non-white vector ($\boldsymbol{x}'$) project it onto the basis along which we want our signal to be sparse, sparsify it and map it back into its original basis. Clearly, if $\kappa = n$ we have $\boldsymbol{x} = \boldsymbol{x}'$ since no clipping takes place.

As a first remark, note that from $\mathbf{E}[\boldsymbol{x}'] = 0$ we have $\mathbf{E}[\boldsymbol{\xi}'] = 0$, $\mathbf{E}[\boldsymbol{\xi}] = 0$ and $\mathbf{E}[\boldsymbol{x}] = 0$. Moreover, if we define the covariance/correlation matrices $\mathscr{X}' = \mathbf{E}[\boldsymbol{x}\boldsymbol{x}'^{\top}]$, $\boldsymbol{\Xi}' = \mathbf{E}[\boldsymbol{\xi}'\boldsymbol{\xi}'^{\top}]$, $\boldsymbol{\Xi} = \mathbf{E}[\boldsymbol{\xi}\boldsymbol{\xi}^{\top}]$, $\mathscr{X} = \mathbf{E}[\boldsymbol{x}\boldsymbol{x}^{\top}]$, we have $\boldsymbol{\Xi}' = \boldsymbol{D}^{\top}\mathscr{X}'\boldsymbol{D}$ and $\mathscr{X} = \boldsymbol{D}\boldsymbol{\Xi}\boldsymbol{D}^{\top}$.

Hence, if $\mathscr{X}'$ is a diagonal matrix both the components of $\boldsymbol{x}'$ and the components of $\boldsymbol{\xi}'$ are independent and the same happens for the nonzero components of $\boldsymbol{\xi} = (\boldsymbol{\xi}')^{\kappa\uparrow}$ causing $\boldsymbol{\Xi}$, and thus also $\mathscr{X}$ to be diagonal. In this case, $\boldsymbol{x}$ will only be κ-sparse but not localized. In fact, by recalling (1.5) we have

$$\mathscr{L}_{\boldsymbol{x}} = \frac{\mathrm{tr}(\mathscr{X}^2)}{\mathrm{tr}^2(\mathscr{X})} - \frac{1}{n} = 0$$

since for diagonal correlations we have $\mathrm{tr}(\mathscr{X}^2) = n\mathscr{X}_{0,0}^2$ while $\mathrm{tr}^2(\mathscr{X}) = n^2\mathscr{X}_{0,0}^2$.

Localization can be imposed by choosing a non-diagonal $\mathscr{X}'$ whose features will approximately be translated into those of $\mathscr{X}$. In fact, since the κ largest components of $\boldsymbol{\xi}'$ are carried over to $\boldsymbol{\xi}$, $\boldsymbol{\xi}$ is the best possible κ-sparse approximation of $\boldsymbol{\xi}'$ and the same relationship holds between $\boldsymbol{x}$ and $\boldsymbol{x}'$. Hence, the larger the κ, the more similar the behavior of $\boldsymbol{x}$ to that of $\boldsymbol{x}'$.

Though, the relationship between the localization of $\mathscr{X}'$ and that of $\mathscr{X}$ is difficult to model analytically we may provide some numerical evidence on the effectiveness of this method within the specific framework that we will use in our examples.

In particular we will consider $\mathscr{X}'$ such that $\mathscr{X}'_{j,k} = \omega^{|j-k|}$ for some $-1 < \omega < 1$. As noted in Chap. 1, this means that $\boldsymbol{x}'$ is a chunk of a stationary stochastic process with power spectrum

$$\Psi(f) = \frac{1-\omega^2}{1+\omega^2-2\omega\cos(2\pi f)}$$

that assumes a high-pass profile for $-1 < \omega < 0$, a flat/white profile for $\omega = 0$, and a low-pass profile for $0 < \omega < 1$. With some calculations one gets

$$\mathscr{L}_{\boldsymbol{x}'} = \frac{2}{n^2}\sum_{j=1}^{n-1} j\omega^{2(n-j)} = \frac{2\omega^2}{n}\frac{n\left(1-\omega^2\right)+\omega^{2n}-1}{n\left(1-\omega^2\right)^2} \tag{2.5}$$

Assume now $n = 128$, and $\boldsymbol{D}$ as the orthonormal Discrete Cosine Transform (DCT) basis. By generating a large amount of sample vectors $\boldsymbol{x}'$ and thus $\boldsymbol{x}$ we may estimate their localization and the power spectrum of the process from which they are taken. The result of such estimations is reported in Figs. 2.5 and 2.6 for different values of ω and sparsity κ (remember that $\kappa = n$ implies $\boldsymbol{x} = \boldsymbol{x}'$, i.e., $\boldsymbol{x}$ is a Gaussian random vector with an exponential correlation controlled by the decay ω).

In particular Fig. 2.5 shows how $\mathscr{L}_{\boldsymbol{x}}$ changes when ω changes. Note that increasing $|\omega|$ increases the localization of the generated signal $\boldsymbol{x}$. As far as *what kind* of localization is conferred to $\boldsymbol{x}$, Fig. 2.6 shows that when $\boldsymbol{x}'$ is low–low pass also $\boldsymbol{x}$ is low-pass, and vice versa.

Overall, our generation methods prove itself to be a practical way of ensuring sparsity while at least qualitatively controlling the localization of the signal and, as such, will be used in all the non-real-world examples of this volume.

To keep such examples not too far from realistic conditions, we refer to Table 1.1 and focus on processes with localizations compatible with those of real-world signals. This helps defining some prototype signals that are reported in Table 2.6.

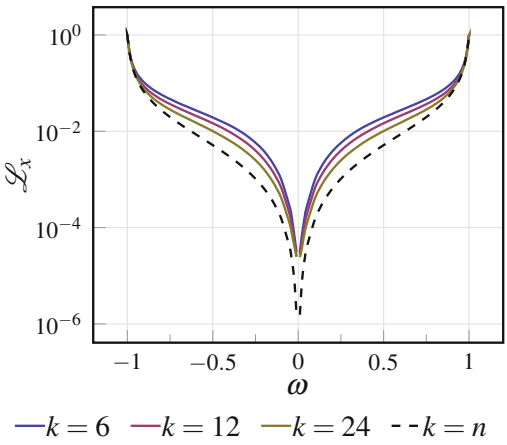

Fig. 2.5 The localization of $\boldsymbol{x}$ when ω changes and for different values of κ

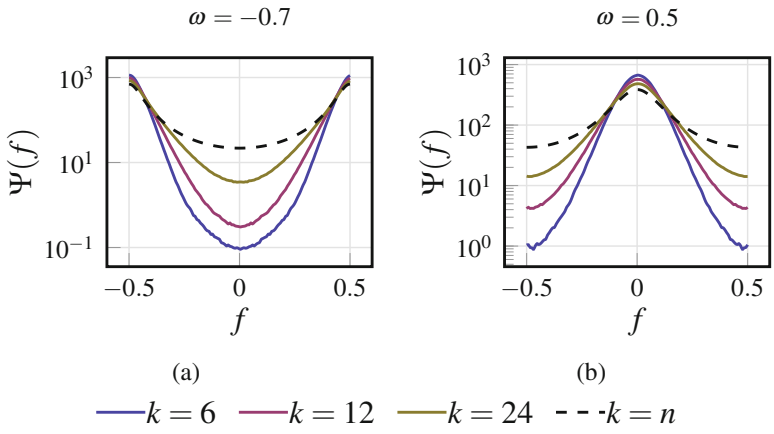

Fig. 2.6 The spectrum of $\boldsymbol{x}$ in a low-pass (**b**) and high-pass (**a**) case for different values of κ

Table 2.6 Definition of prototype signals used in the toy examples

Signal name	$\mathscr{L}_x$	κ	ω
ZL: $\mathscr{L}_x = 0$—white	0	6	0
		12	
		24	
LL: low $\mathscr{L}_x$	0.02	6	± 0.509
		12	± 0.584
		24	± 0.669
ML: medium $\mathscr{L}_x$	0.06	6	± 0.810
		12	± 0.853
		24	± 0.878
HL: high $\mathscr{L}_x$	0.2	6	± 0.959
		12	± 0.964
		24	± 0.966

Since most of our discussion hinges on the design of the matrix $\boldsymbol{A}$ producing the compressed measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, all the examples will address a specific design option or compare a number of them.

To do so we will rely on signals $\boldsymbol{x}$ generated as above and simulate the acquisition process by first perturbing them with a random vector $\boldsymbol{\eta}^{\boldsymbol{x}}$ made of independent, zero-mean Gaussian components whose variance is adjusted to match a prescribed Intrinsic Signal-to-Noise Ratio

$$\texttt{ISNR[dB]} = 20 \log_{10} \left(\frac{\|\boldsymbol{x}\|_2}{\|\boldsymbol{\eta}^{\boldsymbol{x}}\|_2} \right)$$

Such a perturbation is injected to simulate inaccuracies in the acquisition stages, including the possible quantization.

The perturbed signal is then used to produce measurements by using the matrix $\boldsymbol{A}$ under assessment and obtaining $\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{x} + \boldsymbol{\eta}^{\boldsymbol{x}})$. When not explicitly declared otherwise, we will produce the reconstructed signal to be matched against the true signal by feeding $\boldsymbol{y}, \boldsymbol{A}$, and ISNR into the functions provided by the SPGL1 package mentioned in the previous section implementing either the BP or BPDN method, whose robustness and ease of use make it the ideal candidate for the concoction of examples.

2.4 Practical Performance

The first section of this chapter shows that, when not modeled from a worst-case point of view, CS is a promising technique that may allow to reconstruct an n-dimensional signal $\boldsymbol{x}$ from m scalar measurements in a vector $\boldsymbol{y}$ with $m \ll n$.

To give a quantitative appreciation of what can be achieved, assume that $\boldsymbol{x}$ is n-dimensional with $n = 128$, that is $\kappa = 6$ sparse with respect to a DCT orthonormal basis and that is generated as described before with $\omega = 0$ and $\texttt{ISNR[dB]} = 60\,\text{dB}$.

Take $\boldsymbol{A} \sim$ RGE (iid) and, for each value of m from 6 to 64 perform a Montecarlo simulation. For each trial compute the RSNR to accumulate a profile of ARSNR as a function of m. By fixing $\texttt{RSNR[dB]}_{\text{min}} = \texttt{ISNR[dB]} - 5\,\text{dB} = 55\,\text{dB}$, we may also estimate the PCR for each m. The result is reported in Fig. 2.7.

Since both ARSNR and PCR are the-larger-the-better merit figures, the sigmoidal trends in both plots are the practical implication of theorems like Theorem 2.4. In fact, coherently with what theory says, there is some critical value of m after which performance dramatically increases, giving rise to what is often called *phase transition*.

Beyond reflecting theoretical results, plots like those in Fig. 2.7 give a quantitative appreciation of achievable *compression*. For example, Fig. 2.7a shows that for $m = 64$ the ARSNR slightly exceeds the $\texttt{ISNR} = 60\,\text{dB}$ and thus indicates that the original signal, whose dimensionality is $n = 128$, can be acquired with a compression ratio $\texttt{CR} \simeq 2$ with no loss of accuracy (actually with a little amount

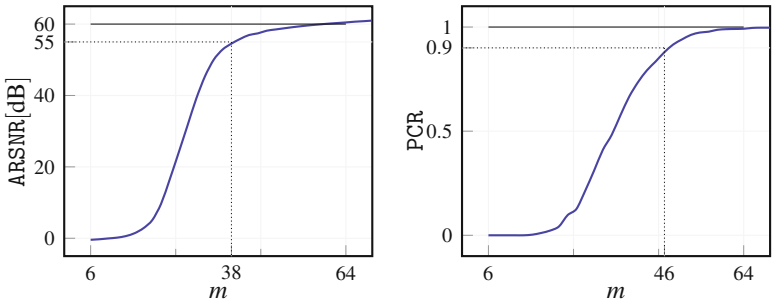

Fig. 2.7 Montecarlo assessment of performance for a classical CS system: when the number of measurements increases both the ARSNR (**a**) and PCR (**b**) increase

of denoising). Yet, one may decide that an $\texttt{ARSNR} = 55\,\text{dB}$ is enough for the application at hand and derive from the same plot that $m = 38$ measurements are enough to meet the specification, increasing the compression ratio to $\texttt{CR} \simeq 3.4$.

Clearly, this concerns average performance. A stricter point of view would be to require that $\texttt{RSNR} = 55\,\text{dB}$ is not achieved on average but at least 90% of the times. Since Fig. 2.7b estimates the probability that RSNR exceeds that threshold as a function of m, one gets that this more stringent specification can be met sizing the system with $m = 46$, that still gives $\texttt{CR} \simeq 2.8$.

This is a somehow impressive performance since it places well apart from the worst-case scenarios that were addressed in deriving the guarantees. As an example, Theorem 1.7, specialized to the case in which $\boldsymbol{x}$ is perfectly κ-sparse with respect to an orthonormal basis, says that the error between the original signal $\boldsymbol{x}$ and its reconstruction $\hat{\boldsymbol{x}}$ can be bounded as

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 = \left\|\boldsymbol{D}\hat{\boldsymbol{\xi}} - \boldsymbol{D}\boldsymbol{\xi}\right\|_2 = \left\|\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}\right\|_2 \leq 4\frac{\sqrt{1+\delta_{2k}}}{1-(\sqrt{2}+1)\delta_{2k}}\epsilon$$

where ϵ is such that $\|\eta\|_2 \leq \epsilon$, and δ_{2k} is the RIC of $\boldsymbol{A}$. For $\delta_{2k} \geq 0$, the coefficient of ϵ is monotonically increasing and thus, even in the best possible conditions, the guarantee of Theorem 1.7 on the RSNR is

$$\texttt{RSNR[dB]} = 20\log_{10}\left(\frac{\|\boldsymbol{x}\|_2}{\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2}\right) \geq 20\log_{10}\left(\frac{\|\boldsymbol{x}\|_2}{4\,\|\boldsymbol{\eta}\|_2}\right) \geq \texttt{ISNR[dB]} - 12\,\text{dB}$$

that, due to its worst-case nature, gives little hint on the fact that, for example, a small average denoising effect can be obtained.

Figure 2.8 shows the trends of same merit figures when the signal to acquire is either $\kappa = 6$-sparse, or $\kappa = 12$-sparse, or $\kappa = 24$-sparse. Clearly, since κ is the minimum number of scalars that are needed to identify $\boldsymbol{x}$, a progressively larger

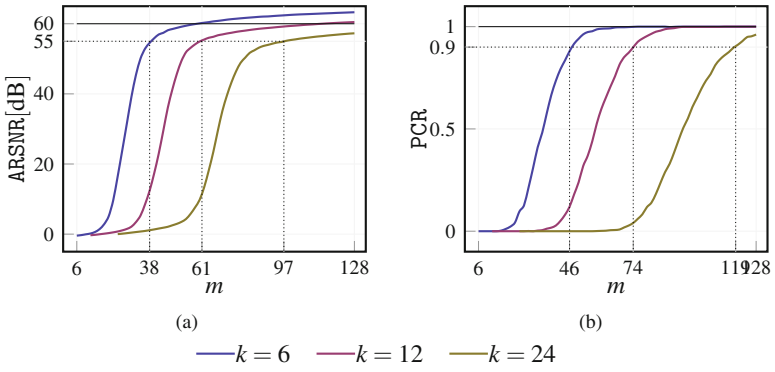

Fig. 2.8 Montecarlo assessment of performance for a classical CS system: when the number of measurements increases both the ARSNR (**a**) and PCR (**b**) increase, though with trends depending on the sparsity κ

Table 2.7 Numerical matching between the asymptotic trend in (1.29) and the empirical evidence of Fig. 2.8. The increase in the sparsity of the signal κ implies an increase in the minimum number m^* of measurements needed to achieve a certain performance that is compared with the $O(\kappa \log(n/\kappa))$ trend

	ARSNR $\geq$ 55 dB		PCR $\geq$ 0.9	
κ	m^*	$\frac{m^*}{\kappa \log_2(n/\kappa)}$	m^*	$\frac{m^*}{\kappa \log_2(n/\kappa)}$
6	38	1.43	46	1.74
12	61	1.49	74	1.81
24	97	1.67	119	2.05

number of measurement is needed to achieve a good signal reconstruction and the corresponding curves move to the right.

As an example, to obtain ARSNR $\geq$ 55 dB one needs at least $m^* = 38$ measurements when the signal is $\kappa = 6$-sparse, but $m^* = 61$ measurements if the signal to reconstruct is $\kappa = 12$-sparse, and $m^* = 97$ measurements for $\kappa = 24$-sparse signals. Though the trend of m^* against n and κ is identified only in asymptotic terms by (1.28) and (1.29), it may be used as a rough estimate of m^* even in finite cases.

In fact, by looking at Table 2.7 one is tempted to adopt as a first sizing criterion $m^* = c\kappa \log_2(n/\kappa)$ with a constant c in the range $2 \leq c \leq 3$.

Though all this may seem a success, from an engineering point of view it is only a starting point. In fact, performances like those in Fig. 2.7 are estimated for a system in which $\boldsymbol{A}$

- has entries that are infinite precision and unbounded;
- is maximally random within the variance constraint on its entries, and thus is completely agnostic both of its role and of its optimization possibilities.

Yet, any real-world implementation of the multiplication of $\boldsymbol{x}$ by $\boldsymbol{A}$ will imply a finite-range calculation with a limited precision, either because of noise if the implementation is analog, or because of quantization if the implementation is digital.

Moreover, instead of simply *accepting* measurements as they happen to be computed by a maximally random policy, one may try to look for measurements that best identify the signal itself so to squeeze as much information as possible in the $m < n$ scalars that will represent $\boldsymbol{x}$. The hope of this quest for *good* measurements is that a smaller number of substantial pieces of information can do the same job of a larger number of purely random looks at the signal.

Leaving the finite-range/finite-precision issue to a following chapter, note that this second aim seems to go against a quite commonly accepted idea, suggestively indicated as *democracy*, that each measurement carries roughly the same amount of information about the signal being acquired. The mathematical foundation of this idea is solid, it has to do with the RIC of the matrices $\boldsymbol{A}$ and with how such constant changes when few rows are dropped: it turns out that when the number of surviving rows is still larger than the minimum number needed to guarantee signal reconstruction, then which row was discarded has little effect on the RIC constant of the resulting matrix.

Yet, the practical effects of such a formal development are negligible for at least two reasons. The first is that we have seen how RIC-based performance bounds are to be taken only as guarantees since they are so loosely correlated with real-world performance that their use as a design criterion is ineffective. In this case, a small change in the RIC of $\boldsymbol{A}$ implies a small change in the performance guarantee but says nothing on the change of the actual performance.

The second is that when seeking for optimally designed CS stages, one never moves from the minimum m needed for a correct reconstruction far enough to be able to speculate about dropping some measurements and still working above that limit. The aim of system optimization is to push m as low as possible.

2.5 Countering the Myth of Democracy and Paving the Way for Practical Optimization

Beyond the considerations in the previous section, the fact that measurement *democracy* is a myth incorrectly inferred from a sound mathematical result can be demonstrated with an easy formal argument if we go back to the simplified setting that characterizes the first sections of this chapter: no noise (i.e., $\texttt{ISNR} = \infty$), and straightforward BP for reconstruction.

Within such a framework, we benefit from a powerful geometric insight on the reasons why the basic reconstruction strategy is successful and when it fails. In particular we may concentrate on failures and have a second look at a slightly rotated and simplified version of Fig. 2.3, that is Fig. 2.9.

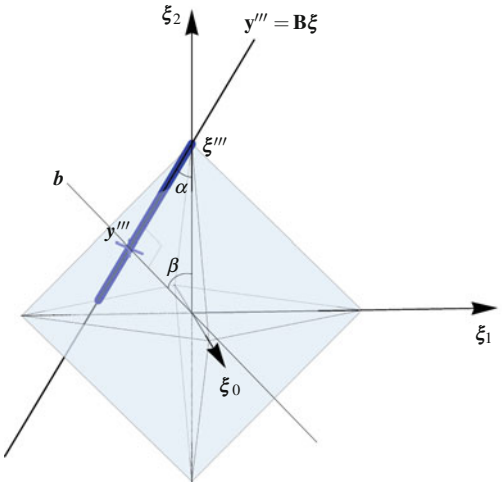

Fig. 2.9 A rotated and simplified version of Fig. 2.3 that highlights the relative positions of the signal $\boldsymbol{\xi}''$, the projection $\boldsymbol{y}'''$ and the face of $S_3^1(\|\boldsymbol{\xi}'''\|_1)$

In that figure, it is easy to verify that the solution to BP is not unique (all the points on the thick blue segment are possible reconstructions of the original signal) due to the fact that the projection plane contains a direction $\boldsymbol{b}$ that is orthogonal to one of the 2-dimensional facets of $S_3^1(\|\boldsymbol{\xi}'''\|_1)$ to which $\boldsymbol{\xi}'''$ itself belongs.

This may be reworded saying that one of the vectors onto which the signal is projected (one of the rows $\boldsymbol{b}$ of the matrix $\boldsymbol{B} = \boldsymbol{AD}$) forms with the signal an angle β such that its complement $\alpha = \pi/2 - \beta$ is equal to the angle between the signal ξ and a facet of $S_3^1(\|\boldsymbol{\xi}'''\|_1)$.

In this case $\alpha = \arccos\left(\sqrt{2/3}\right)$ and if we might ensure that the rows of $\boldsymbol{B}$ avoid forming with the signal an angle $\beta = \pi/2 - \arccos\left(\sqrt{2/3}\right)$, a case like the one depicted in Fig. 2.9 never occurs and signals like ξ''' are correctly reconstructed.

Clearly, other *bad* cases may happen. As an example, Fig. 2.10 shows that there is another choice of $\boldsymbol{b}$ that prevents BP from retrieving the original signal. Again, the reason is that the angle α between the signal and a facet of $S_3^1(\|\boldsymbol{\xi}'''\|_1)$ (in this case it is a 1-dimensional facet) is complementary to the one β between the signal and the direction $\boldsymbol{b}$ along which we are projecting. In this case $\alpha = \beta = \arccos\left(\sqrt{1/2}\right) = \pi/4$.

If we avoided choosing directions $\boldsymbol{b}$ whose angle the signal angles is one of the two computed above, both *bad* cases would be prevented.

Though the detailed proof is out of the scope of this volume, in the general n-dimensional case, when the signal is κ-sparse, angles between $\beta^{\min} = \pi/2 - \arccos(1/\sqrt{1+\kappa})$ and $\beta^{\max} = \pi/2 - \arccos\left(\sqrt{\frac{n-\kappa}{n-\kappa+1}}\right)$ must be avoided. In our case $n = 3$ and $\kappa = 1$ so that $\beta^{\min}$ and $\beta^{\max}$ boil down to $\pi/4$ and $\pi/2 - \arccos\left(\sqrt{2/3}\right)$ computed before. To be on the safe side and not to take too subtle decisions depending on n and κ, all angles in $[\pi/4, \pi/2]$ should be avoided.

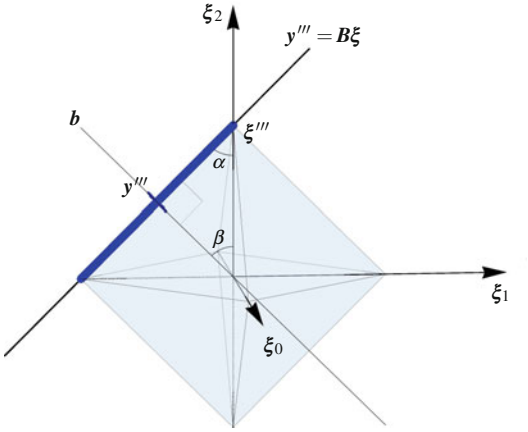

Fig. 2.10 Another *bad* choice of a direction $\boldsymbol{b}$ to use for projection. Also in this case BP cannot reconstruct the original signal as all the points in the *thick blue segment* are possible solutions

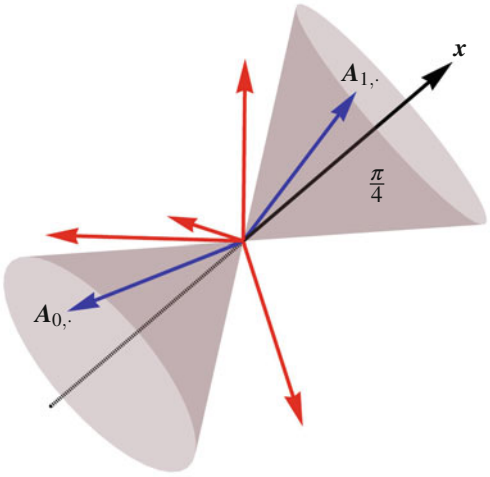

Fig. 2.11 Among many candidate vectors randomly pointing in space, only the two falling in the cone whose axis is $\boldsymbol{x}$ and whose aperture is $\pi/4$ become the rows $\boldsymbol{A}_{0,\cdot}$ and $\boldsymbol{A}_{1,\cdot}$ of the matrix $\boldsymbol{A}$ used for acquisitions

Hence, to ensure maximum reconstruction performance in a noiseless environment, it is advisable to select matrices $\boldsymbol{B}$ whose rows form with ξ an angle strictly smaller than $\pi/4$. When $\boldsymbol{D}$ is an orthonormal basis, this directly translates into a prescription for the angle between the rows of $\boldsymbol{A} = \boldsymbol{B}\boldsymbol{D}^\top$ and the signal $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\xi}$. Such a prescription translates into a simple geometric criterion: rows may be generated as n-dimensional vectors whose entries are independent random variables $\sim \mathrm{N}(0, 1)$ but are included in $\boldsymbol{A}$ only if their angle with $\boldsymbol{x}$ is less than $\pi/4$. Such a method will be indicated as *cone-constrained* CS as accepted rows are vectors falling in the cone whose axis is $\boldsymbol{x}$ and whose aperture is $\pi/4$ as exemplified in Fig. 2.11.

To have a practical appreciation of how much this criterion affects performance we may adopt the same simulation setting as above in the noiseless $\mathtt{ISNR} = \infty$ case and substituting BPDN with BP implemented as a purely linear optimization

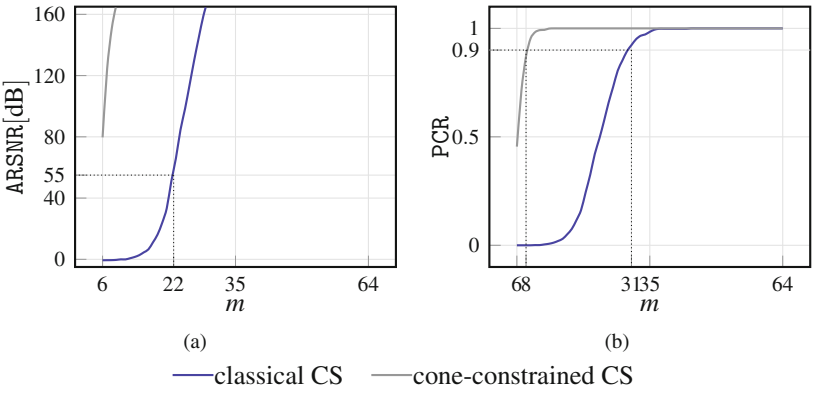

Fig. 2.12 Montecarlo assessment of performance for classical CS (*blue*) and cone-constrained CS (*gray*): the ideal cone-constrained CS clearly exhibits far better performance. (**a**) performance in terms of ARSNR while (**b**) is for PCR

problem (1.32). In these conditions we simulate the performance of classical CS, and cone-constrained CS. The results are shown in Fig. 2.12.

The absence of noise clearly improves reconstruction performance of classical CS. By comparing Fig. 2.7 with Fig. 2.12 we get that an average quality $\texttt{ARSNR} = 55\,\text{dB}$ can be reached with $m = 22$ measurements instead of $m = 38$ measurements, and an $\texttt{RSNR} = 55\,\text{dB}$ can be guaranteed 90% of the times with $m = 31$ measurements instead of $m = 48$ measurements.

Yet, cone-constrained CS has definitely better performance since the average reconstruction quality never falls below 80 dB and $\texttt{RSNR} = 55\,\text{dB}$ can be guaranteed 90% of the times with only $m = 8$ measurements.

Overall, the measurements we select are clearly carrying more information about the signal with respect to measurements picked randomly, and no democracy exists in the real-world. Although this may be taken as a discomforting truth from a social point of view, it is actually extremely good news from the point of view of engineering of CS. In fact, when not all the options are equally good, optimization may be called into play to look for the best design alternatives.

Regrettably, cone-constrained CS is only a theoretical tool since it has no concrete chance to be implemented. To understand why, we have to spend a few words on some very high-level implementation constraints that are at the base of any successful application of CS.

Though it is true that this book focuses on the design of CS stages according to the scheme of Fig. 1.2 we cannot avoid to place such an acquisition subsystem in a slightly more general perspective. This is what Fig. 2.13 does considering the same quantities as in Fig. 1.2. All the acquisition stages (sampling, quantization, compression) can be seen as a single block that *encodes* the analog waveform into the subsufficient-rate sequence of digital scalars $Q(y_k)$. Such sequence is passed to some other subsystem that is interested in knowing $x(t)$ and *decodes* the sequence $Q(y_k)$ into an approximation $\hat{x}(t)$.

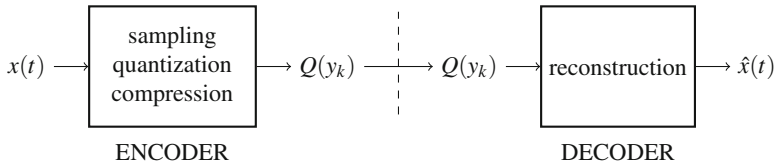

Fig. 2.13 A higher-level view of the role of signal acquisition

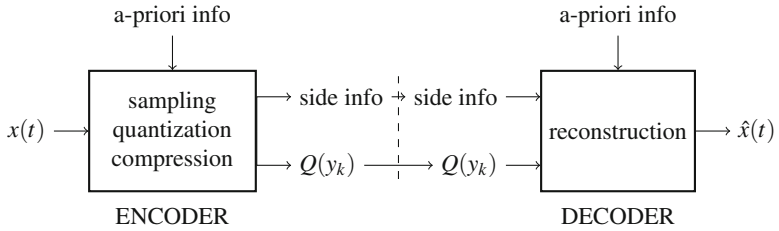

Fig. 2.14 The encoder-channel-decoder view with additional signal paths

The higher-level point of view reveals an encoder–decoder structure that highlights the fact that the only continuous communication between the encoder and the decoder is the subsufficient sequence, i.e., in principle, no other signal dependent information is passed from the acquisition subsystem to the subsystem that uses the acquired signal.

In terms of a CS acquisition mechanism, this means that, for example, the rows of $\boldsymbol{A}$ are not communicated to the decoder, that must be able to know them independently. This is why, a more realistic view at the scheme in Fig. 2.13 should comprise few other details.

First, the encoder and the decoder sides must share some a priori information. If, for example, $\boldsymbol{A}$ is fixed, then it must enter the design of both sides. Alternatively, if $\boldsymbol{A}$ is a time varying instance of a random matrix ensemble (as in our examples), the encoder and the decoder may share the design of a reproducible pseudorandom number generator and the initial state from which it works. In this case the operations of encoder and decoder must be synchronized thus implying a small amount of side information to be transferred from encoder to decoder further to the subsufficient sequence $Q(y_k)$. The resulting more realistic view of the acquisition system is given in Fig. 2.14. Clearly, for the compression scheme to be effective, the total transferred information (the subsufficient sequence plus the side information) must amount to less bits than what would be needed by the sheer transmission of a sufficient sequence of samples.

This is the main reason why cone-constrained CS cannot be effectively employed. In fact, what we may do to apply the method in practice is to deploy two identical copies of a pseudorandom number generator both at the encoder and

the decoder, synchronize them and let them run to produce candidate rows for the matrix $\boldsymbol{A}$. The encoder tests each of them and accepts only the first m of them whose angle with $\boldsymbol{x}$ is less than $\pi/4$ to build $\boldsymbol{A}$. Then, it computes $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ and communicates to the decoder both the vector $\boldsymbol{y}$ and the side information needed to identify the rows it used.

If we assume that to find m rows one must examine M candidates, the number of bits of side information is $\left\lceil \log_2 \binom{M}{m} \right\rceil$ since our task is to identify a specific subset of m elements among M possible candidates. Overall, the amount of information that must be transferred from the encoder to the decoder is $m\mathtt{b}_{\boldsymbol{y}} + \left\lceil \log_2 \binom{M}{m} \right\rceil$, where $\mathtt{b}_{\boldsymbol{y}}$ is the number of bits used for each sample of the subsufficient sequence $Q(y_k)$. This must be compared with the straightforward option of quantizing each samples with $\mathtt{b}_{\boldsymbol{x}}$ bits so that the bitwise compression ratio is

$$\mathrm{CR}^{\mathrm{bit}} = \frac{n\mathtt{b}_{\boldsymbol{x}}}{m\mathtt{b}_{\boldsymbol{y}} + \left\lceil \log_2 \binom{M}{m} \right\rceil} \tag{2.6}$$

Regrettably, the ratio between m and M suffers from a well-known effect of dimensionality on the shape of S_n^2 spheres. Assuming that the candidate rows span all the possible angles uniformly (this is what happens, for example, if their entries are independent normals), the probability that one of them falls within the proper cone is equal to the ratio between the measure of the surface of the two spherical caps $\Gamma' \cup \Gamma''$ illustrated in Fig. 2.15 and the measure of the surface ∂S_n^2 of the whole sphere S_n^2.

From [15] we get that such a ratio is

$$\frac{\mu(\Gamma' \cup \Gamma'')}{\mu(\partial S_n^2)} = B_{\sin^2(\pi/4)}\left(\frac{n-1}{2}, \frac{1}{2}\right) = B_{1/2}\left(\frac{n-1}{2}, \frac{1}{2}\right)$$

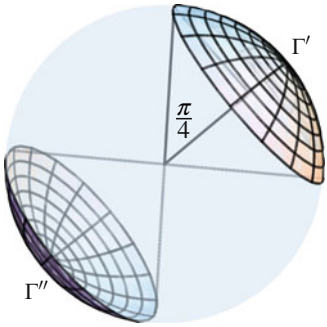

Fig. 2.15 The spherical caps whose surface is proportional to the probability of generating a random measurement falling into the $\pi/4$ cone

that uses the incomplete regularized beta function

$$B_\zeta(p,q) = \frac{\displaystyle\int_0^\zeta t^{p-1}(1-t)^{q-1}\mathrm{d}t}{\displaystyle\int_0^1 t^{p-1}(1-t)^{q-1}\mathrm{d}t}$$

from which we derive that $B_{1/2}\left(\frac{n-1}{2},\frac{1}{2}\right) \leq 2^{-\frac{n-1}{2}}$ for $n \geq 1$ is decreasing not less than exponentially with n. This means, for example, that the probability of a candidate 128-dimensional row of falling into the $\pi/4$ cone whose axis is any given signal $\boldsymbol{x}$ is less than 7.6×10^{-21}.

Assume now that we want to guarantee that $\texttt{RSNR} \geq 55\,\text{dB}$ at least 90% of the times. From Fig. 2.12 we get that $m = 8$ measurements are enough. Yet, the average number of independent candidate rows to evaluate before accumulating $m = 8$ measurements is $8/(7.6 \times 10^{-21}) = 1.1 \times 10^{21}$, and the side information that needs to be communicated amounts to 544 bit. If we assume $\mathtt{b}_y = 12$ (a sensible choice to achieve $\texttt{RSNR} = 55\,\text{dB}$), the total number of bits needed to encode the $n = 128$-dimensional window is $544 + 8 \times 12 = 640$ bit.

Without compression, we may roughly estimate the RSNR achieved by the straightforward quantization of each sample assuming that signal behaves almost sinusoidally so that $\texttt{RSNR} = 6.02\mathtt{b}_{\boldsymbol{x}} + 1.76\,\text{dB}$ where $\mathtt{b}_{\boldsymbol{x}}$ is the number of bits used for each sample. To have $\texttt{RSNR} = 55\,\text{dB}$ we may set $\mathtt{b}_{\boldsymbol{x}} = 9$ so that the total number of bits would be $128 \times 9 = 1152$ bit. Hence, the bitwise compression ratio (2.6) of cone-constrained CS is $\texttt{CR}^{\text{bit}} = 1152/640 \simeq 1.8$.

Note that, if we decided to use every row produced by the generator we would not need to send any side information beyond an initial synchronization, while Fig. 2.12 tells us that the same performance level as before would be guaranteed by $m = 21$ measurements, for a total of $m\mathtt{b}_y = 31 \times 12 = 372$ bit and a corresponding bitwise compression ratio $\texttt{CR}^{\text{bit}} = 1152/372 \simeq 3.1$.

All this said, there is no point in trying an implementation of cone-constrained CS since it does not give any real advantage with respect to purely random CS which also enjoys a much smaller computational burden (even if generating and testing a candidate row took a single nanosecond, accumulating 8 measurements in the $\pi/4$ cone would take more than 33000 years![1]).

However, what we are left with is an intuition that a possible criterion to increase the amount of information that a measurement y carries about the signal $\boldsymbol{x}$ is to obtain it as $y = \boldsymbol{a}^\top\boldsymbol{x}$ using a vector $\boldsymbol{a}$ lying on straight line whose angle with the straight line containing $\boldsymbol{x}$ is 'small', whatever this may signify. From here on what we do can only be intuitively justified but, as we will see in the more applicative

[1]To avoid this curse of dimensionality, the simulations leading to Fig. 2.12 had to generate rows of $\boldsymbol{A}$ by properly modulating the length of random rotations of $\boldsymbol{x}$ itself with angle smaller than $\pi/4$, the conceptual sieving procedure being totally unfeasible.

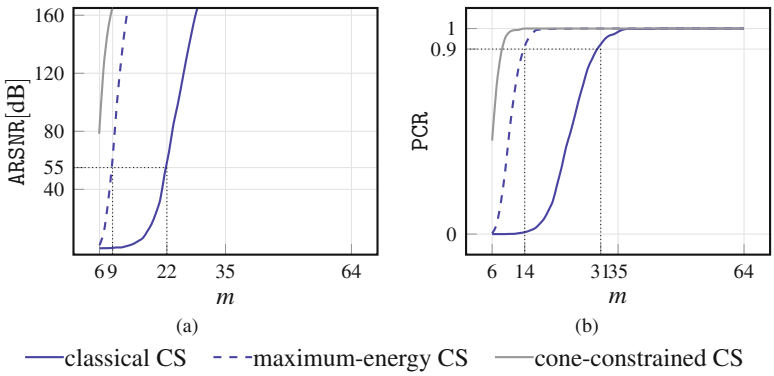

Fig. 2.16 Montecarlo assessment of performance for classical CS (*solid blue*), maximum-energy CS (*dashed blue*), and cone-constrained CS (*solid gray*): maximum-energy CS is not as performing as cone-constrained CS but it outperforms classical CS. (**a**) performance in terms of ARSNR while (**b**) is for PCR

chapters of this volume, gives raise to a powerful heuristic criterion supporting a well-defined and effective design flow for practical CS acquisition.

The first trivial remark is that, given two non-collinear vectors $\boldsymbol{v}'$ and $\boldsymbol{v}'$, forming an angle $\widehat{\boldsymbol{v}' \; \boldsymbol{v}''}$, the angle between the straight lines containing them is $\min\{\widehat{\boldsymbol{v}' \; \boldsymbol{v}''}, \pi - \widehat{\boldsymbol{v}' \; \boldsymbol{v}''}\}$. Hence, such an angle gets smaller when $\widehat{\boldsymbol{v}' \; \boldsymbol{v}''}$ either goes to 0 or π, i.e., when the absolute values $\cos^2\left(\widehat{\boldsymbol{v}' \; \boldsymbol{v}''}\right)$ increases. Since $y^2 = \left(\boldsymbol{a}^\top \boldsymbol{x}\right)^2 = \|\boldsymbol{a}\|_2^2 \, \|\boldsymbol{x}\|_2^2 \cos^2\left(\widehat{\boldsymbol{a} \; \boldsymbol{x}}\right)$ and $\boldsymbol{x}$ is assigned, if we may assume that all the rows of $\boldsymbol{A}$ have approximately the same length, smaller angles correspond to higher energies of the measurement y.

A new, heuristic method is naturally born from these considerations. In a system analogous to what has been sketched for cone-constrained CS, let the row generator produce M candidates. Then compose the matrix $\boldsymbol{A}$ with the rows $\boldsymbol{a}$ corresponding to the m largest values of $(\boldsymbol{a}^\top \boldsymbol{x})^2$. The measurements are computed as $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ and passed to the decoder. This strategy is named *maximum-energy* CS.

In this new configuration both M and m are degrees of freedom. This gives us some control on the amount of bits spent on side information $\left\lceil \log_2 \binom{M}{m} \right\rceil$, an amount that must be traded with the quality of the reconstruction. In this case we do not have a theoretical background allowing to anticipate reconstruction performance and we have to rely on simulations. If we do so, we may add a track to Fig. 2.12 and obtain Fig. 2.16.

Maximum-energy CS is simulated generating $M = 512$ candidates and taking the m largest energy measurements for $m = \kappa = 6$ to $m = n/2 = 64$. Since it is only a heuristic approximation of the cone-constrained policy, performance decreases but is still much higher than that of classical CS.

If we assume that our target reconstruction quality is $\texttt{ARSNR} \geq 55\,\text{dB}$, then classical CS achieves it with $m = 22$ while maximum-energy CS requires only $m = 9$. These numbers allow to compute the bitwise compression ratio (2.6) for the same case as above in which $\texttt{b}_x = 9$ and $\texttt{b}_y = 12$. Classical CS yields $\texttt{CR}^{\text{bit}} = 1152/(22 \times 12) = 1152/264 \simeq 4.4$. Maximum-energy CS yields $\texttt{CR}^{\text{bit}} = 1152/\left(9 \times 12 + \left\lceil \log_2 \binom{512}{9} \right\rceil\right) = 1152/171 \simeq 6.7$ and is therefore able to provide a gain further to the mere reduction of the number of scalar measurements.

If we assume that our target reconstruction quality is to guarantee that 90% of the times we have $\texttt{RSNR} \geq 55\,\text{dB}$, then classical CS achieves it with $m = 31$ while maximum-energy CS requires only $m = 14$. The corresponding bitwise compression ratios are $\texttt{CR}^{\text{bit}} \simeq 3.1$ for classical CS and $\texttt{CR}^{\text{bit}} \simeq 4.5$ for maximum-energy CS.

Overall, maximum-energy CS seems to be a good candidate to leverage the intuitive criterion we have developed for measurement quality while keeping adaptivity to a level that can be managed by adding a reasonable amount of side information.

This is true even in a noisy environment. In fact, we may go back to our original setting in which $\texttt{ISNR} = 60\,\text{dB}$, and keep the same configuration of maximum-energy CS to obtain curves like the ones in Fig. 2.17.

Since noise is back, performance deteriorates also for maximum-energy CS. Yet, the new method is still able to yield better bitwise compression ratios. In fact, looking at Fig. 2.17a we get that the reference quality level $\texttt{ARSNR} = 55\,\text{dB}$ is achieved with $m = 38$ measurements by classical CS, and with $m = 16$ measurements by maximum-energy CS. The usual computation of $\texttt{CR}^{\text{bit}}$ with $\texttt{b}_x = 9$ and $\texttt{b}_y = 12$ gives $\texttt{CR}^{\text{bit}} = 1152/456 \simeq 2.5$ for classical CS and $\texttt{CR}^{\text{bit}} = 1152/292 \simeq 3.9$ for maximum-energy CS.

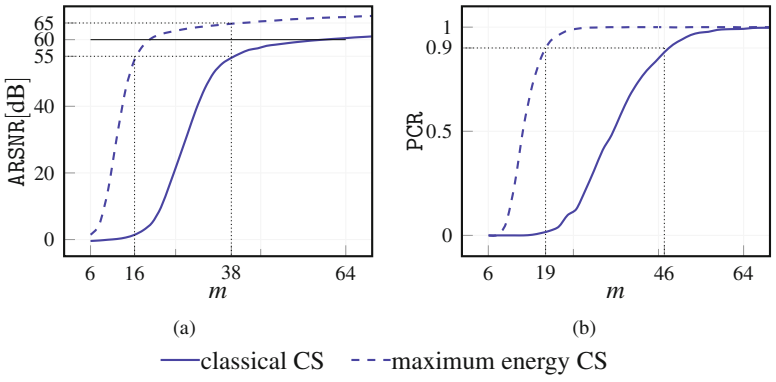

Fig. 2.17 Montecarlo comparison between performance of classical CS (*solid*) and maximum-energy CS (*dashed*): both the ARSNR (**a**) and PCR (**b**) curves are dramatically improved

Figure 2.17a also shows that the adapted method is able to provide noteworthy average denoising. For example, when classical CS achieves our reference performance level $\texttt{ARSNR} = 55\,\text{dB}$, maximum-energy CS is able to yield $\texttt{ARSNR} = 65\,\text{dB} > \texttt{ISNR}$. Clearly, this comes at some expense since maximum-energy CS accumulates and communicates $\left\lceil \log_2 \binom{512}{38} \right\rceil = 192\,\text{bit}$ in addition to the 456 bit used to encode the measurements. Yet, in this case reconstruction provides an accuracy that would not be attained by simply encoding the samples.

To confirm that what we have observed so far can be of use, we may explore different types of signals with different sparsities. As before, no formal analysis is available but an extensive Montecarlo assessment can be pursued using different values of κ and different values of ω in the exponential correlation signal mode we defined in Sect. 2.3.

A sample of the results of such an assessment is reported in Fig. 2.18 where we give the trends of both ARSNR and PCR (with target $\texttt{RSNR} = 55\,\text{dB}$) when $n = 128$ and $\kappa \in \{6, 12, 24\}$ while trying a high-pass signal with medium localization (ML HP), a white signal, and a low-pass signal with large localization (HL LP) as defined in Table 2.6. The white case with $\kappa = 6$ is the same as the one in Fig. 2.17.

In terms of the minimum number of measurements needed to match a certain reconstruction quality, the improvement of maximum-energy CS over classical CS is undoubtable. The fact that it may result in a better bitwise compression ratio must be checked case by case. From Fig. 2.18 it is clear that the performance of classical CS is independent of localization while that of maximum-energy CS is not. Hence, we may focus on the white case and summarize our quantifications in Table 2.8.

Overall, the maximum-energy criterion seems to behave rather well. Yet, its implementation has two drawbacks that may limit practical application. First, one has to compute many more measurements ($M = 512$ in our examples) than what is then communicated to the decoder. If the cost of computing a measurement is not negligible, this may have an impact on the encoder complexity. Due to the dimensionality effect that we already noted when analyzing cone-constrained CS, such an impact is expected to dramatically increase as n increases. Moreover, side information must be computed and communicated to the decoder further to measurements and this may also imply an increased encoder complexity.

Intuitively, this potentially increased complexity depends on the fact that maximum-energy CS automatically adapt itself to the specific instance $\boldsymbol{x}$ of the signal it is acquiring.

In the next chapter, we will see that a slightly less performing method can be devised which, leveraging the same measurement energy principle and adapting to the class of signals to acquire rather than to a specific instance, allows to increase performance while keeping encoder complexity to a minimum.

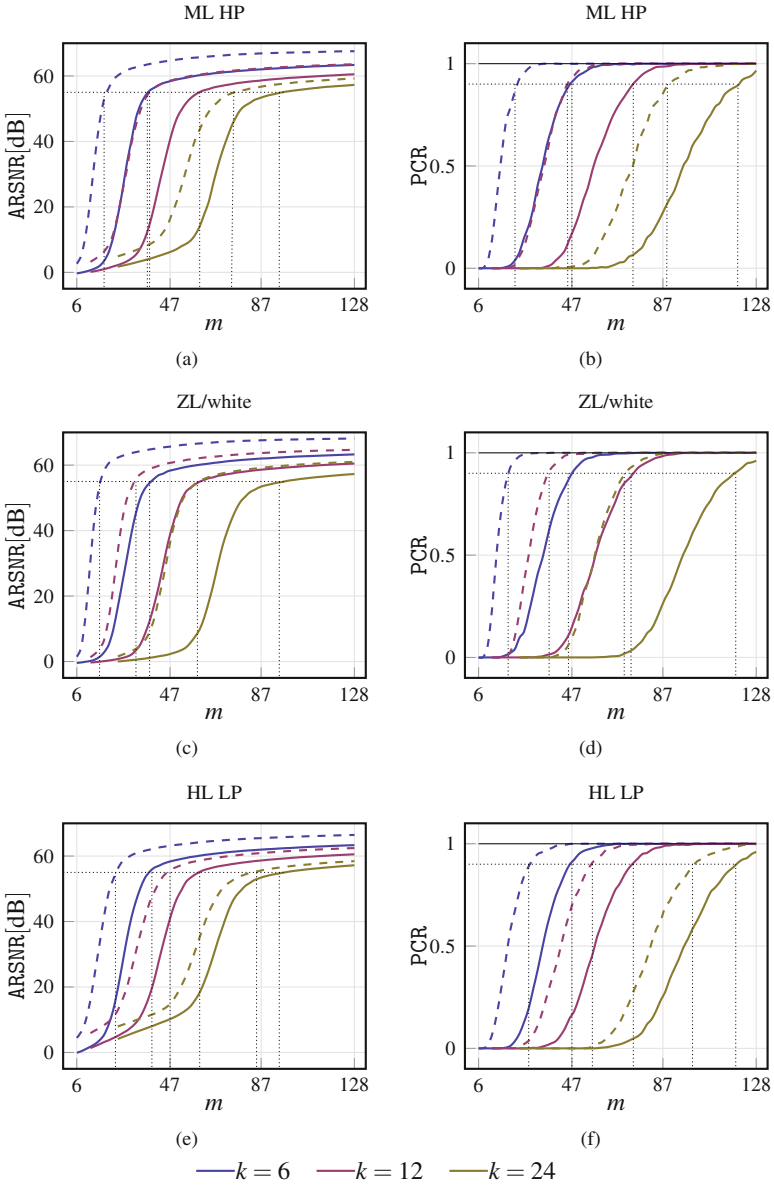

Fig. 2.18 Montecarlo comparison between performance of classical CS (*solid*) and maximum-energy CS (*dashed*): both the ARSNR (**a**) and PCR (**b**) curves are dramatically improved in all configurations

Table 2.8 Bitwise compression ratios of classical CS and maximum-energy CS when $\mathtt{b}_x = 8$ and $\mathtt{b}_y = 12$ and for two different reconstruction quality requirements. Straightforward encoding of $n = 128$ samples would require 1152 bit

ARSNR = 55 dB							
	Maximum-energy CS				Classical CS		
κ	m	M	$m\mathtt{b}_y + \lceil \log_2 \binom{M}{m} \rceil$	$\mathtt{CR}^{\mathrm{bit}}$	m	$m\mathtt{b}_y$	$\mathtt{CR}^{\mathrm{bit}}$
6	16	512	292	3.9	38	452	2.5
12	32	512	553	2.1	59	708	1.6
24	59	512	968	1.2	95	1140	1.0
PCR = 0.9							
	Maximum-energy CS				Classical CS		
κ	m	M	$m\mathtt{b}_y + \lceil \log_2 \binom{M}{m} \rceil$	$\mathtt{CR}^{\mathrm{bit}}$	m	$m\mathtt{b}_y$	$\mathtt{CR}^{\mathrm{bit}}$
6	19	512	342	3.9	47	564	2.5
12	37	512	632	1.8	73	876	1.3
24	70	512	1131	1.0	119	1428	0.81

References

1. S. Becker, J. Bobin, E.J. Candès, NESTA: a fast and accurate first-order method for sparse recovery. SIAM J. Imag. Sci. **4**(1), 1–39 (2011)
2. E. van den Berg, M.P Friedlander, Probing the Pareto frontier for basis pursuit solutions. SIAM J. Sci. Comput. **31**(2), 890–912 (2008)
3. T. Blumensath, M.E. Davies, Iterative thresholding for sparse approximations. J. Fourier Anal. Appl. **14**(5-6), 629–654 (2008)
4. D. Bortolotti et al., Energy-aware bio-signal compressed sensing reconstruction on the WBSN-gateway. IEEE Trans. Emerg. Top. Comput. **PP**(99), 1–1 (2016)
5. P.L. Combettes, J.-C. Pesquet, A proximal decomposition method for solving convex variational inverse problems. Inverse Prob. **24**(6), p. 065014 (2008)
6. I. Daubechies et al., Iteratively reweighted least squares minimization for sparse recovery. Commun. Pure Appl. Math. **63**(1), 1–38 (2010)
7. D.L. Donoho, Neighborly Polytopes and Sparse Solution of Underdetermined Linear Equations, Technical report, Department of Statistics, Stanford University, 2005
8. D.L. Donoho, J. Tanner, Counting faces of randomly projected polytopes when the projection radically lowers dimension. J. Am. Math. Soc. **22**(1), 1–53 (2009)
9. D.L. Donoho, J. Tanner, Observed universality of phase tran- sitions in high-dimensional geometry with implications for modern data analysis and signal processing. Philos. Trans. R. Soc. Lond. A Math. Phys. Eng. Sci. **367**(1906), 4273–4293 (2009)
10. D.L. Donoho, J. Tanner, Precise undersampling theorems. Proc. IEEE **98**(6), 913–924 (2010)
11. I.F. Gorodnitsky, B.D. Rao, Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. IEEE Trans. Signal Process. **45**(3), 600–616 (1997)
12. M. Grant, S. Boyd, CVX: Matlab Software for Disciplined Convex Programming version 2.1. http://cvxr.com/cvx, Mar 2015
13. M. Grant, S. Boyd, Graph implementations for nonsmooth convex programs, in *Recent Advances in Learning and Control*, ed. by V. Blondel, S. Boyd, H. Kimura. Lecture Notes in Control and Information Sciences (Springer, Heidelberg, 2008), pp. 95–110
14. S. Ji, Y. Xue, L. Carin, Bayesian compressive sensing. IEEE Trans. Signal Process. **56**(6), 2346–2356 (2008)

15. S. Li, Concise formulas for the area and volume of a hyperspherical cap. Asian J. Math. Stat. **4**(1), 66–70 (2011)
16. D. Needell, J.A. Tropp, CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. Appl. Comput. Harmon. Anal. **26**(3), 301–321 (2009)
17. S. Rangan, Generalized approximate message passing for estimation with random linear mixing, in *2011 IEEE International Symposium on Information Theory Proceedings*, IEEE, July 2011, pp. 2168–2172

http://www.springer.com/978-3-319-61372-7

Adapted Compressed Sensing for Effective Hardware Implementations
A Design Flow for Signal-Level Optimization of Compressed Sensing Stages

Mangia, M.; Pareschi, F.; Cambareri, V.; Rovatti, R.; Setti, G.

2018, XIV, 319 p. 180 illus., 142 illus. in color.,
Hardcover
ISBN: 978-3-319-61372-7