Chapter 2

Compressive Sensing

In this chapter we will outline the main ideas of the *Compressive Sensing* framework (also known as *Compressed Sensing, Compressed Sampling, Compressive Samping* or *CS*). We begin by discussing conventional approaches to data acquisition and compression, and how CS differs from that. Next, formulate the CS Problem. Following that, we briefly discuss the general solution strategy. Lastly, give a brief overview of deterministic approaches to CS reconstruction (maybe).

In Chapter 5, we will discuss a Bayesian approach to solving the Compressive Sensing Problem.

2.1 Signal Acquisition and Compression

2.1.1 Acquisition

In order to work with information within analog signals (continuous streams of data) such as sounds, images or video, we rely on reducing the analog signals to digital (discrete) signals that can be processed with computers. This digitization is done by taking discrete measurements of the analog signal at certain points in time or space, a process known as *sampling*.

Conventional approaches to sampling are based on the *Shannon/Nyquist Sampling Theorem* [3]: When sampling a band-limited signal uniformly, we are able to *perfectly reconstruct* the signal from its samples if the sampling rate is at least twice the bandwidth of the signal.

We briefly illustrate this in Figure 2.1. Consider an analog signal $x(t)$ that varies with time, such as an audio wave. Let f be the highest frequency present in $x(t)$.

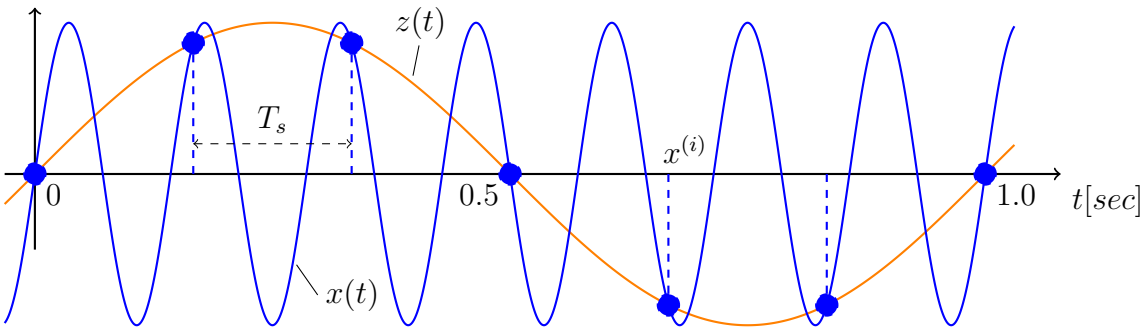

Fig. 2.1 Illustration of the Shannon/Nyquist Sampling Theorem. The orange curve is the original signal $x(t)$ which is a sinusoid with frequency $f = 7$ Hz. The blue points are discrete samples $x^{(i)}$ taken from $x(t)$ at a sampling rate $f_s = 7$ Hz, which is below the Nyquist Rate $2f = 14$ Hz. Thus, aliasing occurs and interpolation algorithms will reconstruct an alias $z(t)$ of $x(t)$.

In order to digitise $x(t)$, we measure x at discrete points in time $t^{(0)}, \cdots, t^{(n)}$ and store the samples $x^{(i)} \equiv x(t^{(i)})$. We sample x uniformly, measuring a sample every T_s seconds, so that $t^{(i)} = iT_s$. The sampling rate is therefore $f_s = 1/T_s$.

Suppose we wish to reconstruct $x(t)$ by interpolating the samples. There is an infinite number of continuous functions that fit this set of samples. However, it can be shown that only one of them has a bandwidth of no more than $f_s/2$. Thus, if $f < f_s/2$ (the *Nyquist Criterion*), then $x(t)$ is the unique function that will be approximated by interpolation algorithm such as the *Whittaker-Shannon interpolation formula*[3].

In Figure 2.1, we have a sinusoidal signal $x(t)$ with frequency f. The sampling rate is $f_s = f$ and therefore below the signal's *Nyquist rate* $2f$. Thus, we are unable to reconstruct $x(t)$ from the samples. Instead, we will reconstruct an *alias* $z(t)$ which, in this case, is another sinusoid with frequency $f/7$. The original signal x is lost.

We have illustrated Nyquist sampling in the 1-dimensional case. The same principles hold for higher dimensional signals such as images and videos.

For signals that vary with space, the sampling rate is governed by the desired spatial resolution. In order to recover the finer details (the high-frequency components) of an image, we require higher pixel density (i.e. more pixels per

Nyquist sampling underlies almost all signal acquisition protocols that are found in practice. It is the basis of medical imaging devices, consumer electronics such as audio and video recorders, radio receivers, etc.

2.1.2 Compression

The sampling theorem imposes a lower bound on the sampling rate above which we get are able to perfectly reconstruct the desired signal. This lower bound is often very high and we end up with a very large number of measurements. Storage and transfer of such signals becomes prohibitavely expensive as the size of the signal grows. Thus, a need for *data compression* arises.

We will discuss a particular type data compression known as *transform coding*. It is the standard compression method for "natural" and manmade signals such as audio, photos, and video and is the basis of many common signal formats such as JPEG for images, MPEG for videos and MP3 for audio.

Let $\boldsymbol{v}$ by a real-valued digital signal of length M, $\boldsymbol{v} \in \mathbb{R}^M$. Without loss of generality, $\boldsymbol{v}$ is assumed to be a one-dimensional signals. If we are working with a multi-dimensional signals, we may first vectorize it into a long vector. When compressing digital signals, we are usually interested in *lossy compression*.

Any vector in $\mathbb{R}^M$ can be expressed as a linear combination of M *basis vectors* $\boldsymbol{\psi}_j \in \mathbb{R}^M$:

$$\boldsymbol{v} = \sum_{j=1}^{M} w_j \boldsymbol{\psi}_j \tag{2.1}$$

where w_j is the coefficient (or weight) associated with $\boldsymbol{\psi}_j$.

By forming the *basis matrix* $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1 \cdots \boldsymbol{\psi}_M]$, we can express equation (2.1) in matrix form

$$\boldsymbol{v} = \boldsymbol{\Psi}\boldsymbol{w}$$

where $\boldsymbol{w} = (w_1, \cdots, w_M)^T$. For simplicity, we assume that the basis $\boldsymbol{\Psi}$ is orthonormal, so that $\boldsymbol{\Psi}\boldsymbol{\Psi}^T = \boldsymbol{I}_M$ and $\boldsymbol{\psi}_i^T \boldsymbol{\psi}_j$ is 1 if $i = j$ and 0 otherwise. Thus, the coefficient w_j is given by $w_j = \boldsymbol{v}^T \boldsymbol{\psi}_j$.

We now have two equivalent representations of the same signal, $\boldsymbol{v}$ in the original basis and $\boldsymbol{w}$ in the $\boldsymbol{\Psi}$ basis. Since $\boldsymbol{\Psi}$ is orthogonal, $\boldsymbol{v}$ and $\boldsymbol{w}$ have the same ℓ_2-norm, $||\boldsymbol{v}||_2 = ||\boldsymbol{\Psi}\boldsymbol{w}||_2 = ||\boldsymbol{w}||_2$.

While in the original signal $\boldsymbol{v}$ the energy is spread over many of its components, it is possible to find a basis $\boldsymbol{\Psi}$ such that the energy of the transformed signal $\boldsymbol{w}$ is concentrated in only a few large components w_j. Thus a large fraction of the entries in $\boldsymbol{w}$ are very close to zero.

Suppose that we delete the entries w_j that are very small and replace them with zero to obtain $\hat{\boldsymbol{w}}$. Let $\hat{\boldsymbol{v}} = \boldsymbol{\Psi}\hat{\boldsymbol{w}}$ the approximate signal in the original domain. Since

(a) Uncompressed Image (b) Compressed Image via DCT

Fig. 2.2 The uncompressed image has a resolution of 512×512, i.e. 262144 pixels. We compress the image by performing a Discrete Cosine Transform and storing only the largest 27832 coefficients. The compression ratio is 9.42.

$\hat{\boldsymbol{w}}$ is very close to $\boldsymbol{w}$, so that $||\hat{\boldsymbol{w}} - \boldsymbol{w}||_2$ is very small, it follows that

$$||\hat{\boldsymbol{v}} - \boldsymbol{v}||_2 = ||\Psi\hat{\boldsymbol{w}} - \Psi\boldsymbol{w}||_2 = ||\Psi(\hat{\boldsymbol{w}} - \boldsymbol{w}||_2 = ||\hat{\boldsymbol{w}} - \boldsymbol{w})||_2$$

is also very small.

Thus, a viable method for lossy compression of the signal $\boldsymbol{v} \in \mathbb{R}^M$ would be the following:

1. Compute the full set of transform coefficients $\{w_j\}_{j=1}^M$ via $\boldsymbol{w} = \Psi^T\boldsymbol{v}$.
2. Locate all the coefficients w_j whose absolute value is above a certain threshold (suppose there are K of them).
3. Discard all the $(M - K)$ small coefficients
4. Store the values and locations of the K large coefficients

In order to view the compressed signal in the original domain, we perform the transform: $\boldsymbol{\Psi}\hat{\boldsymbol{w}} = \hat{\boldsymbol{v}}$, where $\hat{\boldsymbol{w}}$ is $\boldsymbol{w}$ with the $(M - K)$ smallest coefficients replaced by zero.

It is possible to find basis matrices $\boldsymbol{\Psi}$ that result in very high compression ratios for a wide range of natural signals without any noticable reduction in the signal quality. Furthermore, many of the commonly used basis transforms can be computed very efficiently.

Audio signals and a wide class of communication signals are highly compressible in the localized Fourier basis. Images and video signals, on the other hand, can often be compressed via the *Discrete Cosine Transform* (DCT) or the *Discrete Wavelet Transform* (DWT). For instance, the JPEG standard for image compression is based on the DCT, while the more modern JPEG2000 format uses the CDF 9/7 wavelet transform or the CDF 5/3 wavelet transform [5].

In Figure 2.2, we compress the standard test image "Lenna" via a DCT. We are only storing about 10% of the transform coefficients. Yet, the difference between the original image and the compressed image is hardly noticable.

We will discuss the DCT and the DWT in more detail in Chapter 3.

2.2 A Novel Approach: Compressive Sensing

The conventional approach to data acquisition and compression is very effective and has been highly influential. However, it is also extremely wasteful. We acquire a huge amount of data at the signal sampling stage and then proceed to discard a large part of it at the compression stage.

Compressive Sensing is a more general approach that lets us *acquire signals directly in a compressed format*. This is clearly more efficient as it allows us to skip the intermediate stage of taking N samples.

In this section we will formulate the Compressive Sensing problem. For simplicity, the focus will be on discrete signals such as digital images or videos.

2.2.1 Problem Formulation

Let $\boldsymbol{v} \in \mathbb{R}^M$ be the signal of interest. As an example, $\boldsymbol{v}$ could be a digital photograph, such as Figure 2.2a, that has been unrolled into a long vector of length M, where M is the number of pixels.

Suppose $\boldsymbol{v}$ is currently unknown and we want to acquire a compressed representation of it without measuring $\boldsymbol{v}$ directly. To do so, consider the following linear sensing scheme: We measure inner products between the signal $\boldsymbol{v}$ and a collection of M-dimensional vectors $\{\boldsymbol{\theta}_i\}_{i=1}^N$ to obtain the measurements $y_j = \boldsymbol{v}^T\boldsymbol{\theta}_i$ for $i = 1, \cdots, N$. This relation can be expressed more succinctly as

$$\boldsymbol{y} = \boldsymbol{\Theta}\boldsymbol{v} \tag{2.2}$$

where $\boldsymbol{y} = (y_1, \cdots, y_N)^T$ and $\boldsymbol{\Theta}$ is the $N \times M$ *sensing matrix* given by

$$\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\theta}_1^T \\ \vdots \\ \boldsymbol{\theta}_N^T \end{bmatrix}. \tag{2.3}$$

We are interested in the *undersampled* situation where $N < M$. In particular, we would like to acquire a compressed representation $\boldsymbol{y}$ of $\boldsymbol{v}$ using as few measurements as possible, while still being able to recover the original signal.

There are two problems that stand out at this point. First, given that $N << M$, how can we ensure that, during the measurement process, we do not lose any information contained in $\boldsymbol{v}$. Second, given our measurements $\boldsymbol{y}$ and the sensing matrix $\boldsymbol{\Theta}$, how do we recover the signal of interest $\boldsymbol{v}$.

Recovery of $\boldsymbol{v}$ corresponds to solving the underdetermined linear system (2.2). In general, this is i

2.2.2 Sparsity

The crucial concept that enables us to address these problems is sparsity.

References

[1] DeVore, R. A., Jawerth, B., and Lucier, B. J. (1992). Image compression through wavelet transform coding. *Information Theory, IEEE Transactions on*, 38(2):719–746.

[2] Pilikos, G. (2014). Signal reconstruction using compressive sensing. MPhil thesis, University of Cambridge.

[3] Shannon, C. E. (1949). Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21.

[4] Stollnitz, E. J., DeRose, T. D., and Salesin, D. H. (1995). Wavelets for computer graphics: a primer. 1. *Computer Graphics and Applications, IEEE*, 15(3):76–84.

[5] Taubman, D. and Marcellin, M. (2012). *JPEG2000 Image Compression Fundamentals, Standards and Practice: Image Compression Fundamentals, Standards and Practice*. The Springer International Series in Engineering and Computer Science. Springer US.

[6] Tipping, A. and Faul, A. (2002). Analysis of sparse bayesian learning. *Advances in neural information processing systems*, 14:383–389.

[7] Tipping, M. E. (2001). Sparse bayesian learning and the relevance vector machine. *The journal of machine learning research*, 1:211–244.

[8] Tipping, M. E., Faul, A. C., et al. (2003). Fast marginal likelihood maximisation for sparse bayesian models. In *AISTATS*.