Sparse deconvolution (an MM algorithm)

Ivan Selesnick
Polytechnic Institute of New York University
selesi@poly.edu
October 7, 2012
Last edit: October 21, 2014

1 Introduction

Deconvolution refers to the problem of estimating the unknown input to an LTI system when the output signal and system response are known. In practice, the available output signal is also noisy. For some systems, the deconvolution problem is quite straight forward; however, for systems that are non-invertible or nearly non-invertible (e.g. narrow-band or with frequency response nulls), the problem is more difficult. The use of an exact inverse system can greatly amplify the noise rendering the result useless. In such cases, it is important to utilize prior knowledge regarding the input signal so as to obtain a more accurate estimate of the input signal, even when the system is nearly non-invertible and the observed output signal is noisy.

In some applications of deconvolution, it is known that the input signal is *sparse* (i.e. a spike train, etc.) or approximately sparse. Applications of 'sparse deconvolution' include geophysics, ultrasonic non-destructive evaluation, speech processing, and astronomy [11]. One approach to sparse deconvolution involves the minimization of a cost function defined in terms of the ℓ_1 norm [4, 6, 16]. The minimization of cost functions defined in terms of the ℓ_1 norm is useful not just for deconvolution, but for sparse signal processing more generally. Indeed, since its early application in geophysics, the ℓ_1 norm and sparsity have become important tools in signal processing [3]. The tutorial [14] compares least squares and ℓ_1 norm solutions for several signal processing problems, illustrating the advantages of a sparse signal model (when valid).

This tutorial aims to illustrate some of the principles and algorithms of sparse signal processing, by way of considering the sparse deconvolution problem. A computationally efficient iterative algorithm for sparse deconvolution is derived using the majorization-minimization (MM) optimization method. The MM method is a simple, yet effective and widely applicable, method that replaces a difficult minimization problem with a sequence of simpler ones [8]. Other algorithms, developed for general ℓ_1 norm minimization, can also be used here [5, 13, 17]. However, the MM-derived algorithm takes advantage of the banded structure of the matrices arising in the sparse deconvolution problem. The resulting algorithm uses fast solvers for banded linear systems [1], [12, Sect 2.4].

The conditions that characterize the optimal solution are described and illustrated in Sec. 3. With these simple conditions, the optimality of the result computed by a numerical algorithm can be readily verified. Moreover, as described in Sec. 4, the optimality conditions provide a straight forward way to set the regularization parameter λ based on the noise variance (if it is known).

1.1 Problem

We assume the noisy data $y(n)$ is of the form

$$y(n) = (h * x)(n) + w(n), \tag{1}$$

where $h(n)$ is the impulse response of an LTI system, $x(n)$ is a sparse signal, $w(n)$ is white Gaussian noise, and '$*$' denotes convolution. We assume here that the LTI system can be described by a recursive difference equation,

$$y(n) = \sum_k b(k)\, x(n-k) - \sum_k a(k)\, y(n-k) \tag{2}$$

where $x(n)$ is the input signal and $y(n)$ is the output signal.

$$x(n) \quad \longrightarrow \boxed{h(n)} \longrightarrow \quad y(n) = (h * x)(n)$$

Note that the difference equation (2) can be written in matrix form as

$$\mathbf{A}\mathbf{y} = \mathbf{B}\mathbf{x} \tag{3}$$

where $\mathbf{A}$ and $\mathbf{B}$ are banded matrices. For example, if the difference equation is first order,

$$y(n) = b(0)\, x(n) + b(1)\, x(n-1) - a(1)\, y(n-1),$$

then $\mathbf{A}$ and $\mathbf{B}$ have the form

$$\mathbf{A} = \begin{bmatrix} 1 & & & & \\ a_1 & 1 & & & \\ & \ddots & \ddots & & \\ & & a_1 & 1 & \\ & & & a_1 & 1 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} b_0 & & & & \\ b_1 & b_0 & & & \\ & \ddots & \ddots & & \\ & & b_1 & b_0 & \\ & & & b_1 & b_0 \end{bmatrix}. \tag{4}$$

From (3), the output $\mathbf{y}$ of the system can be written as

$$\mathbf{y} = \mathbf{A}^{-1}\mathbf{B}\mathbf{x} = \mathbf{H}\mathbf{x}$$

where the system matrix $\mathbf{H}$ is

$$\boxed{\mathbf{H} = \mathbf{A}^{-1}\mathbf{B}}$$

Note that, even though $\mathbf{A}$ and $\mathbf{B}$ are banded matrices, $\mathbf{H}$ is not. (The inverse of a banded matrix is not banded in general.) The data model (1) is written in matrix form as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}.$$

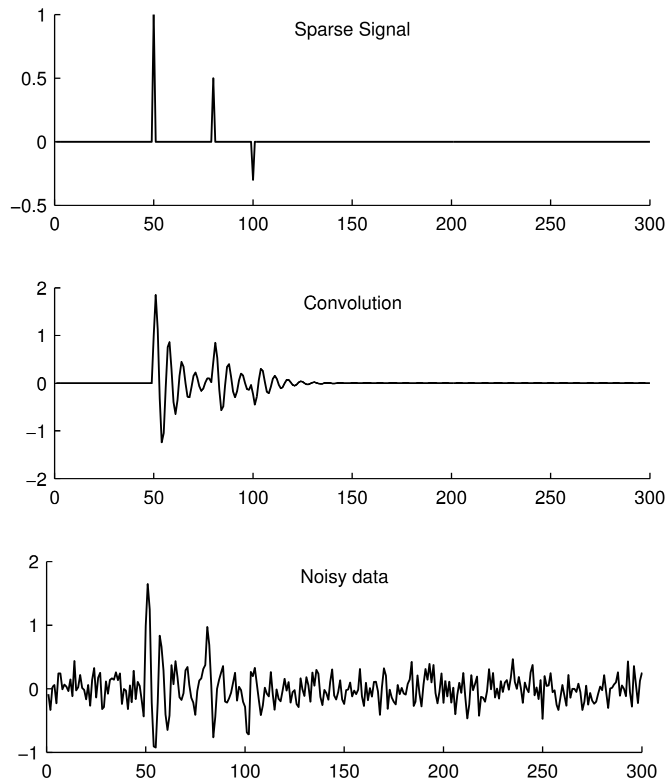


Figure 1: Sparse signal $x(n)$, output of convolution system, and observed data $y(n)$.

Example. An example is illustrated in Fig. 1. The sparse signal $x(n)$

$$x(n) = \delta(n-50) + 0.5\,\delta(n-80) - 0.3\,\delta(n-100)$$

is the input to a convolution system. The system is second order, defined by the difference equation coefficients:

$$\begin{aligned} & b_0 = 1, \quad b_1 = 0.8, \\ & a_0 = 1, \quad a_1 = -2r\cos(\omega), \quad a_2 = r^2 \\ & \text{where } \ \omega = 0.95, \quad r = 0.9 \end{aligned} \tag{5}$$

The system has complex poles at $r\mathrm{e}^{\pm \mathrm{j}\omega}$. The output $(h * x)(n)$ exhibits decaying oscillations. The output signal is then corrupted by additive white Gaussian noise with standard deviation $\sigma = 0.2$.

1.2 Optimization formulation

In order to estimate $x(n)$ from the data $y(n)$, we consider the optimization problem

$$\boxed{\arg\min_{\mathbf{x}} \ \frac{1}{2}\,\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1} \tag{6}$$

The use of the ℓ_1 norm as the regularization term captures the prior knowledge (or model) that $x(n)$ is sparse. The ℓ_1 norm is not the only way to measure sparsity, however, among convex functionals, it is the most natural. The regularization parameter $\lambda > 0$ should be chosen according to the level of the noise $w(n)$. A method for setting λ is described in Sec. 4.

Problem (6) is useful for many sparse signal processing problems, not just deconvolution. In the general case, problem (6) is called 'basis pursuit denoising' (BPD) [5] or the 'lasso' [17]. References [5, 17] give examples (other than deconvolution) of problems posed in this form, and motivations for using the ℓ_1 norm.

1.3 Notation

The N-point signal x is represented by the vector

$$\mathbf{x} = [x(0), \ldots, x(N-1)]^T.$$

The ℓ_1 norm of a vector $\mathbf{v}$ is defined as

$$\|\mathbf{v}\|_1 = \sum_n |v(n)|.$$

The ℓ_2 norm of a vector $\mathbf{v}$ is defined as

$$\|\mathbf{v}\|_2 = \Big[\sum_n |v(n)|^2\Big]^{\frac{1}{2}}.$$

2 Minimizing the cost function

The optimization problem (6) can not be solved directly — it is not differentiable. However, the cost function is convex and therefore the theory and practice of convex optimization can be brought to the problem. In these notes we describe an approach for solving (6), suitable whenever $\mathbf{H}$ has the form $\mathbf{A}^{-1}\mathbf{B}$ where $\mathbf{A}$ and $\mathbf{B}$ are both banded. The algorithm described below converges in practice in few iterations and is computationally efficient. It is based on the majoriziation-minimization optimization method and exploits the availability of fast algorithms for solving banded systems of linear equations [12, Sect 2.4]. The derivation below closely follows that in Ref. [8].

2.1 Majorization-Minimization

This section briefly describes the majorization-minimization (MM) approach [8] for minimizing a convex cost function $F(\mathbf{x})$. To minimize $F(\mathbf{x})$, the majorization-minimization approach solves a sequence of simpler optimization problems:

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} G_k(\mathbf{x}) \tag{7}$$

where k is the iteration counter, $k = 0, 1, 2, \ldots$. The function $G_k(\mathbf{x})$ must be chosen as a majorizer of $F(\mathbf{x})$, i.e.

$$G_k(\mathbf{x}) \geq F(\mathbf{x}), \quad \forall \mathbf{x}$$
$$G_k(\mathbf{x}^{(k)}) = F(\mathbf{x}^{(k)})$$

The majorizer should be chosen so as to be relatively easy to minimize. With initialization $\mathbf{x}_0$, the update (7) produces a sequence of vectors $\mathbf{x}^{(k)}$, $k \geq 1$, converging to the minimizer of $F(\mathbf{x})$. For more details see Ref. [8] and references therein.

In these notes, we will use a majorizer for the ℓ_1 norm. Note that

$$\boxed{\frac{1}{2}\mathbf{x}^T \mathbf{\Lambda}\, \mathbf{x} + \frac{1}{2}\|\mathbf{v}\|_1 \geq \|\mathbf{x}\|_1, \qquad [\mathbf{\Lambda}]_{n,n} = \frac{1}{|v(n)|}} \tag{8}$$

with equality when $\mathbf{x} = \mathbf{v}$. Therefore, the left-hand-side of (8) is a majorizer of $\|\mathbf{x}\|_1$. The matrix $\mathbf{\Lambda}$ is a diagonal matrix with elements of $1/|\mathbf{v}|$ on the diagonal. The derivation of this majorizer is illustrated in more detail in Ref. [15] where it is used to derive an algorithm for total variation denoising.

2.2 Iterative algorithm

We will minimize the cost function

$$F(\mathbf{x}) = \frac{1}{2}\,\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1 \tag{9}$$

using majorization-minimization (MM). A majorizer of $F(\mathbf{x})$ can be obtained by majorizing just the ℓ_1 norm using (8). In this way, a majorizer of $F(\mathbf{x})$ is given by

$$G_k(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \frac{1}{2}\mathbf{x}^T \mathbf{\Lambda}_k\, \mathbf{x} + \frac{\lambda}{2}\|\mathbf{x}^{(k)}\|_1$$

where matrix $\mathbf{\Lambda}_k$ is a diagonal matrix with elements of $\lambda/|\mathbf{x}^{(k)}|$ on the diagonal.

$$[\mathbf{\Lambda}_k]_{n,n} = \frac{\lambda}{|x^{(k)}(n)|} \tag{10}$$

Therefore, the MM update (7) for $\mathbf{x}^{(k)}$ is

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} \left[\frac{1}{2}\,\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \frac{1}{2}\mathbf{x}^T \mathbf{\Lambda}_k\, \mathbf{x}\right]. \tag{11}$$

The last term of $G_k(\mathbf{x})$ has been omitted because it does not depend on $\mathbf{x}$. Note that (11) is quadratic in $\mathbf{x}$ so the minimizer can be written using linear algebra. The solution to (11) can be written explicitly as

$$\mathbf{x}^{(k+1)} = \Big(\mathbf{H}^T\mathbf{H} + \mathbf{\Lambda}_k\Big)^{-1}\mathbf{H}^T\mathbf{y} \tag{12}$$

or in terms of $\mathbf{A}$ and $\mathbf{B}$, as

$$\mathbf{x}^{(k+1)} = \Big(\mathbf{B}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{B} + \mathbf{\Lambda}_k\Big)^{-1}\mathbf{B}^T\mathbf{A}^{-T}\mathbf{y}. \tag{13}$$

Although (13) is mathematically valid, there are two problems with this update. First, as elements of $\mathbf{x}$ go to zero, elements of $\mathbf{\Lambda}_k$ go to infinity; so the update (13) may become numerically inaccurate. (Note that, due to sparsity of $\mathbf{x}$, it is expected and intended that elements of $\mathbf{x}$ do in fact go to zero!) Second, the update (13) calls for the solution to a large system of linear equations which has a high computational cost. Moreover, the system matrix is not banded, because $(\mathbf{A}\mathbf{A}^T)^{-1}$ is not banded; so fast solvers for banded systems can not be used here. Both issues are by-passed by using the matrix inverse lemma as suggested in Ref. [9].

Using the matrix inverse lemma, the matrix inverse in (13) can be rewritten as

$$\Big(\underbrace{\mathbf{B}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{B}+\mathbf{\Lambda}_k}_{\text{not banded ...}}\Big)^{-1} = \mathbf{\Lambda}_k^{-1} - \mathbf{\Lambda}_k^{-1}\mathbf{B}^T\Big(\underbrace{\mathbf{A}\mathbf{A}^T+\mathbf{B}\mathbf{\Lambda}_k^{-1}\mathbf{B}^T}_{\text{banded!}}\Big)^{-1}\mathbf{B}\mathbf{\Lambda}_k^{-1} \tag{14}$$

The indicated matrix is banded because $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{\Lambda}$ are all banded. For convenience, we define $\mathbf{W}_k = \mathbf{\Lambda}_k^{-1}$, i.e.,

$$[\mathbf{W}_k]_{n,n} = \frac{|x^{(k)}(n)|}{\lambda}. \tag{15}$$

Then, we can write

$$\Big(\underbrace{\mathbf{B}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{B}+\mathbf{\Lambda}_k}_{\text{not banded ...}}\Big)^{-1} = \mathbf{W}_k - \mathbf{W}_k\mathbf{B}^T\Big(\underbrace{\mathbf{A}\mathbf{A}^T+\mathbf{B}\mathbf{W}_k\mathbf{B}^T}_{\text{banded!}}\Big)^{-1}\mathbf{B}\mathbf{W}_k. \tag{16}$$

Using (16), the update (13) can be implemented as

$$\mathbf{g} \leftarrow \mathbf{B}^T\mathbf{A}^{-T}\mathbf{y} \tag{17}$$

$$\mathbf{W}_k \leftarrow \frac{1}{\lambda}\mathrm{diag}(|\mathbf{x}^{(k)}|) \tag{18}$$

$$\mathbf{x}^{(k+1)} \leftarrow \mathbf{W}_k\Big[\mathbf{g} - \mathbf{B}^T\Big(\mathbf{A}\mathbf{A}^T+\mathbf{B}\mathbf{W}_k\mathbf{B}^T\Big)^{-1}\mathbf{B}\mathbf{W}_k\mathbf{g}\Big] \tag{19}$$

Equations (18) and (19) constitute the loop of the algorithm. Note that all matrices arising in the update equations are banded, hence the matrix computations can be implemented with fast high efficiency low-memory algorithms [1]. Also note that $\mathbf{W}_k$, not $\mathbf{\Lambda}_k$, appears in the update equations, so the problem of division by zero (or by very small numbers) does not arise.

The update equations (17)-(19) constitute an algorithm for solving (6). It is only necessary to initialize $\mathbf{x}_0$ and set $k = 1$. Suitable initializations for $\mathbf{x}_0$ include either $\mathbf{x}_0 = \mathbf{y}$ or $\mathbf{x}_0 = \mathbf{H}^T\mathbf{y}$.

A MATLAB program `deconvL1` implementing sparse deconvolution using this algorithm is given in Sec. 7. The program uses sparse matrix structures so that MATLAB calls fast solvers for banded systems [1] by default.

2.3 Example

For the data shown in Fig. 1, the sparse deconvolution solution is shown in Fig. 2. The solution was obtained using the MM algorithm described above. Due to the properties of MM algorithms, the cost function is monotonically decreasing, as reflected by the cost function history shown in Fig. 2. For this example, 100 iterations of the MATLAB program `deconvL1` took 54 msec on a MacBook (2.4 GHz Intel Core 2 Duo) running MATLAB version 7.8.

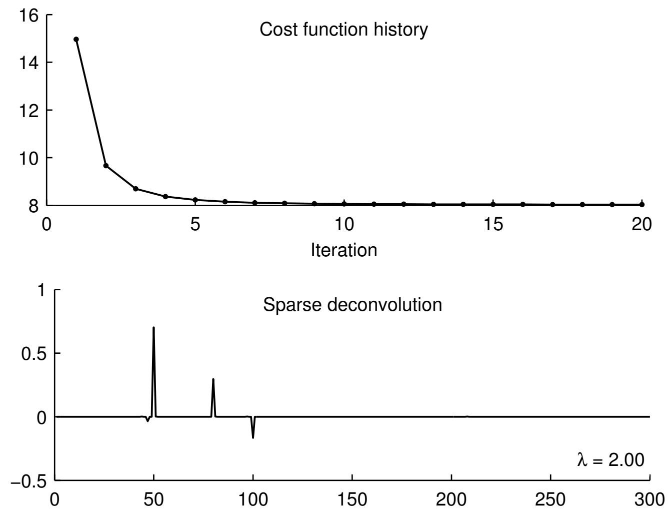


Figure 2: Sparse deconvolution of the data shown in Fig. 1. The deconvolution is performed by ℓ_1 norm minimization using the MM algorithm.

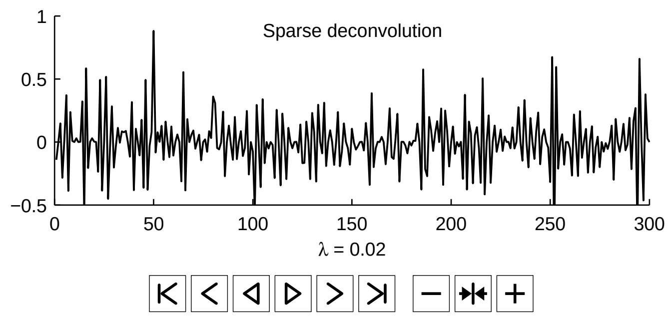


Figure 3: Animation of sparse deconvolution as a function of the regularization parameter λ. The animation shows the effect of λ on the solution. (Animation requires Adobe Reader.)

Note that the solution depends on λ. Fig. 3 is an animation that shows the sparse deconvolution as a function of λ.[1] Each frame of the animation shows the solution for a different value of λ. The animation illustrates how the solution depends on the regularization parameter λ. For small λ, the solution is very noisy; for large λ, the solution is overly attenuated. The question of how to set λ will be discussed in Section 4.

Least squares. It is informative to compare sparse deconvolution with least squares deconvolution. A least squares version of problem (6) is:

$$\arg\min_{\mathbf{x}} \; \frac{1}{2}\, \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \frac{\lambda}{2}\, \|\mathbf{x}\|_2^2 \tag{20}$$

which has the solution

$$\mathbf{x}_{\text{LS}} = \left(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I}\right)^{-1}\mathbf{H}^T\mathbf{y}.$$

[1] The animation requires Adobe Reader. (At the time of writing, other pdf viewers will not show the animation.)

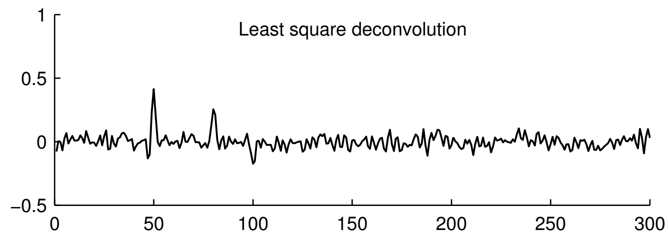


Figure 4: Least squares deconvolution.

Like the ℓ_1 norm solution, this can be rewritten using the matrix inverse lemma so as to exploit the computational efficiency of fast solvers for banded systems.

Fig. 4 shows the least square solution for a particular value of λ. Compared to the sparse deconvolution solution, the least square solution is both noisier and attenuated. Reducing λ leads to an even noisier solution. Increasing λ leads to further attenuation of the solution. The sparse solution is superior to the least square solution here, because in this example, the sparse signal model is valid.

3 Optimality conditions

It turns out that the solution to the ℓ_1 norm optimization problem (6) must satisfy conditions that can be readily checked [2]. Using these conditions, it can be verified if the obtained result really is the optimal solution. In addition, the optimality conditions lead to a simple approach for setting the regularization parameter, λ, to be discussed in Sec. 4.

If $\mathbf{x}$ minimizes (6), then it must satisfy:

$$|\mathbf{H}^T(\mathbf{y} - \mathbf{H}\mathbf{x})| \leq \lambda \tag{21}$$

where the absolute value $|\cdot|$ is taken element-wise.

For the sparse signal $\mathbf{x}$ illustrated above in Fig. 2, the vector $\mathbf{H}^T(\mathbf{y} - \mathbf{H}\mathbf{x})$ is illustrated in Fig. 5a. The dashed line in Fig. 5a is λ. The figure shows that $\mathbf{x}$ does indeed satisfy (21).

Condition (21) by itself does not guarantee that $\mathbf{x}$ is the solution to (6). It is necessary but not sufficient.

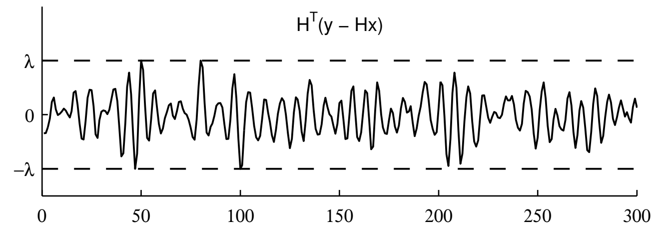
Define

$$\mathbf{g} = \mathbf{H}^T(\mathbf{y} - \mathbf{H}\mathbf{x}).$$

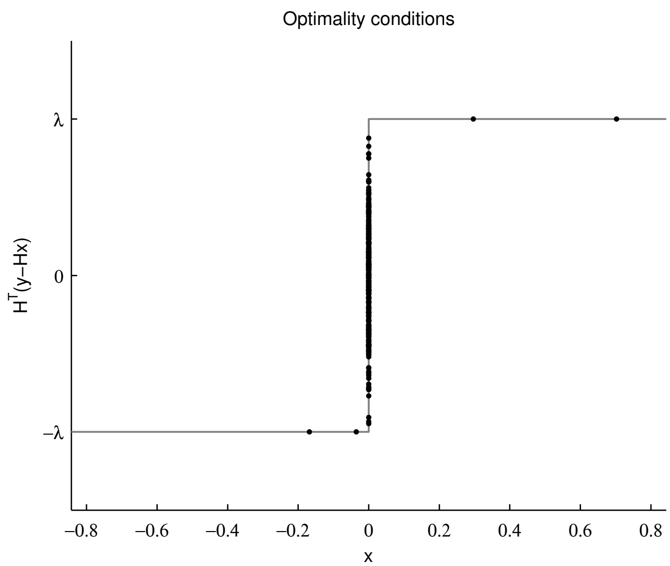
Then $\mathbf{x}$ minimizes (6) if and only if

$$\begin{aligned} g(n) &= \lambda, && \text{for } x(n) > 0 \\ g(n) &= -\lambda, && \text{for } x(n) < 0 \\ |g(n)| &\leq \lambda, && \text{for } x(n) = 0. \end{aligned} \tag{22}$$

This condition is illustrated in Fig. 5b as a scatter plot. Each point in the plot represents a pair $(x(n), g(n))$. It can be seen that each time sample $(x(n), g(n))$ must lie on the graph of the sign



(a) The signal $\mathbf{g} = \mathbf{H}^T(\mathbf{y} - \mathbf{H}\mathbf{x})$ is bounded by λ.



(b) Scatter plot of $g(n)$ versus $x(n)$. The points lie on graph of the sign function.

Figure 5: Optimality conditions. The solution to (6) must satisfy the constraints (22).

function. The sparsity of $\mathbf{x}$ can be recognized by the fact that most of the points lie on the line given by $x = 0$.

The convergence of the MM algorithm to the solution of (6) can be observed by monitoring how closely $\mathbf{x}^{(k)}$ satisfies the optimality conditions (22). Figure 6 is an animation of the scatter plot $(x(n), g(n))$ as the algorithm progresses through its iterations.[2] Each frame shows a different iteration of the algorithm. As the algorithm progresses, the points in the scatter plot approach the graph of the sign function.

4 Setting the regularization parameter, λ

Note that setting λ larger (more regularization) yields an estimate with less noise, but more signal distortion (attenuation). Setting λ smaller (less regularization) yields an estimate with more noise.

[2]The animation requires Adobe Reader or Adobe Acrobat. (At the time of writing, other pdf viewers will not show the animation.)

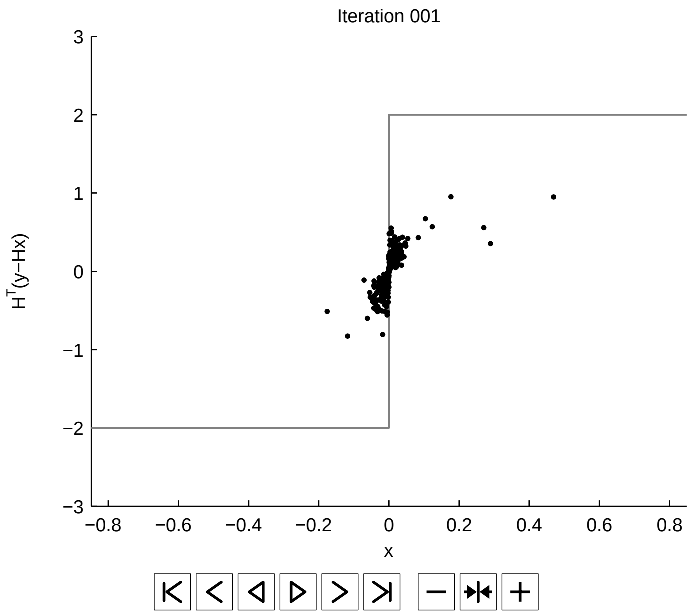


Figure 6: Animation of convergence of the MM algorithm. The points in the scatter plot approach the optimality constraints. (Animation requires Adobe Reader.)

How can a suitable value of λ be found based on the known convolution system and noise statistics?

The optimality conditions (22) suggest a simple procedure to set the regularization parameter, λ. First, note that if λ is sufficiently large, i.e.

$$\lambda \geq \max(|\mathbf{H}^T\mathbf{y}|),$$

then the solution to (6) will be be identically zero, $\mathbf{x} \equiv 0$. This follows from (22). For $\mathbf{x} \equiv 0$ to be a solution, according to (22), $\lambda \geq |g(n)|$. When $\mathbf{x} \equiv 0$, then $\mathbf{g} = \mathbf{H}^T\mathbf{y}$.

One approach to set λ is based on considering the case $\mathbf{x} \equiv 0$. When $\mathbf{x} \equiv 0$, then the data $\mathbf{y}$ is simply the noise signal $\mathbf{w}$. To ensure that the solution in this case is indeed identically zero, λ should be chosen so that

$$\lambda \geq \max|\mathbf{H}^T\mathbf{w}|. \tag{23}$$

In practice, the noise $\mathbf{w}$ is not known, but its statistics may be known. In many cases, it is modeled as zero-mean white Gaussian with known variance σ^2. In this case $\mathbf{H}^T\mathbf{w}$ will be zero-mean Gaussian with variance $\sigma^2 \cdot \sum_n |h(n)|^2$ where $h(n)$ is the impulse response of the convolution system. While (23) asks that λ be greater than the largest value of the sequence $\mathbf{H}^T\mathbf{w}$, this is not directly applicable when $\mathbf{w}$ is a Gaussian random vector, because its pdf is not finitely supported, i.e. the maximum is not finite. However, a practical implementation of (23) is obtained by replacing $\max|\mathbf{H}^T\mathbf{w}|$ by, say, three times the standard deviation of $\mathbf{H}^T\mathbf{w}$. According to the 'three-sigma rule', a zero-mean Gaussian random variable rarely exceeds three times its standard deviation. Hence, we get the rule

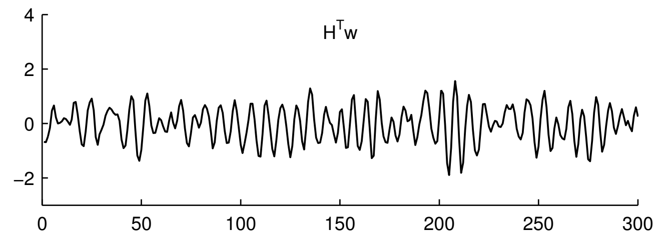


Figure 7: Noise analysis for setting λ.

$$\lambda \geq 3\sigma \sqrt{\sum_n |h(n)|^2}. \tag{24}$$

Note that using a value λ larger than necessary will lead to more attenuation of the signal, therefore, we should use the *smallest* value λ that effectively eliminates the noise. Hence, we should set λ *equal* to the value shown in (24).

For the system (5) and $\sigma = 0.2$ as used in the example illustrated in Fig. 1, the rule (24) gives $\lambda = 2.01$. In the sparse deconvolution result shown in Fig. 2, $\lambda = 2$ was used. It is informative to note that the signal $\mathbf{H}^T\mathbf{w}$, shown in Fig. 7, has a maximum absolute value of 1.85. The value of λ, 2.01, provided by the '3-sigma rule', is quite close to this value. This illustrates the effectiveness of the rule.

If the noise $\mathbf{w}$ is not white, or is non-Gaussian, then the same concept can be adapted to obtain a suitable value for λ. The result will depend on the autcorrelation function or other properties of the noise, depending on what is known.

5 Example: Sparse deconvolution of speech

As a second example, we apply sparse deconvolution to a segment of voiced speech, illustrated in Fig. 8. The speech waveform is eight milliseconds in duration, with a sampling rate of $F_s = 7418$ samples/second.[3]

It is common to model voiced speech $y(n)$ as the output of a filter driven by a spiky pulse-train $x(n)$,

$$x(n) \longrightarrow \boxed{H(z)} \longrightarrow y(n)$$

where the filter $H(z)$ is an all-pole filter (AR filter). The filter $H(z)$ is usually obtained by using AR modeling applied to a short segment of speech. Each segment of speech is modeled using a different filter; so the filter is effectively time-varying.

Here we use an AR model of order 12 to model the speech waveform. The all-pole filter, obtained from the speech waveform, is shown in Fig. 8. Both the impulse response and pole-zero diagram

[3]The speech waveform is an extract of a person saying 'Matlab'. In MATLAB: `load mtlb; [b,a] = butter(2, 0.05, 'high'); y = filtfilt(b, a, mtlb); y = y(1000:1600);` The high-pass filter removes baseline drift.

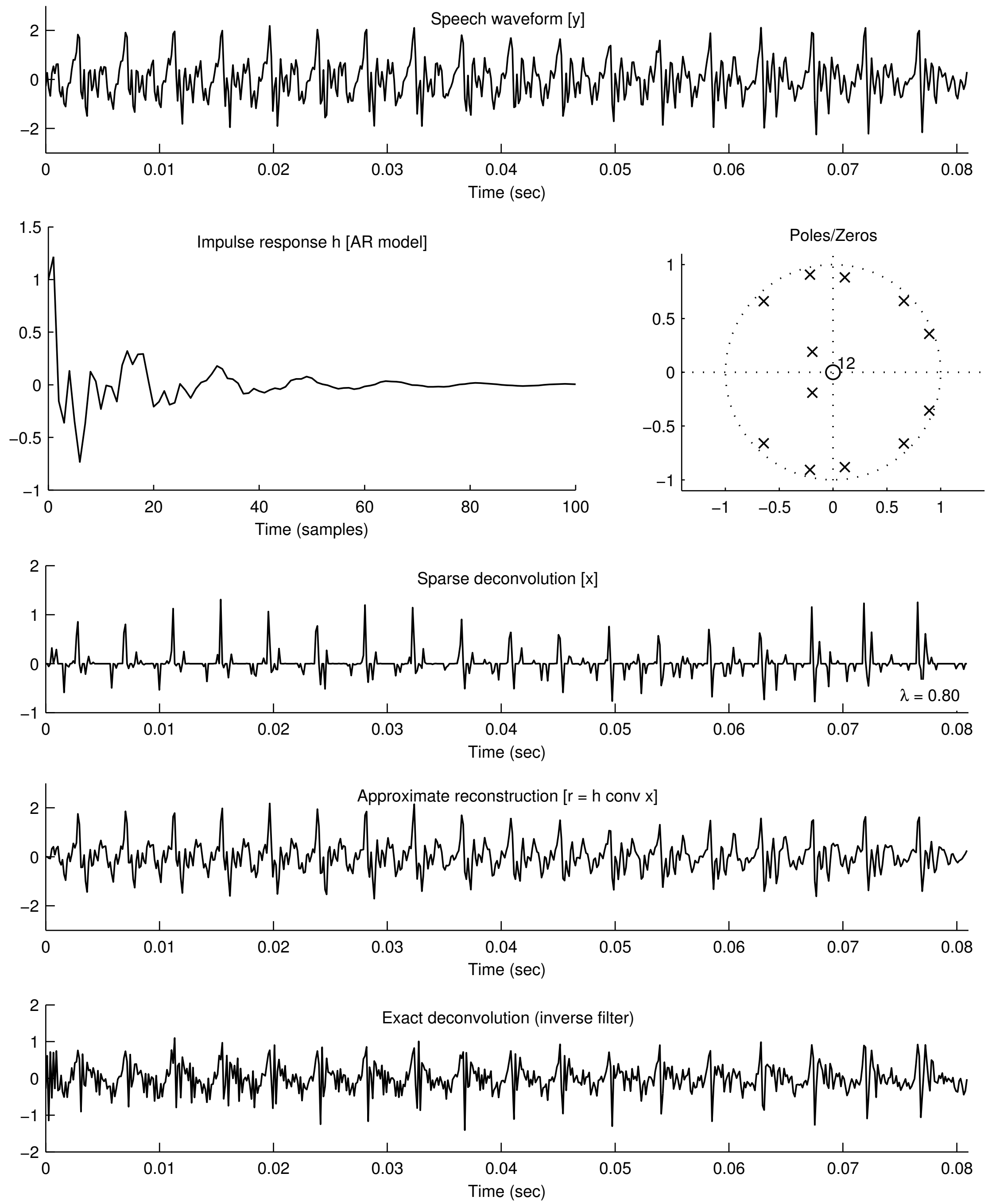


Figure 8: AR modeling and sparse deconvolution applied to a segment of voiced speech.

are shown. Instead of conventional AR modeling, sparse linear prediction can be used to obtain the filter with improved results for speech analysis and coding [10]. Here we have used conventional AR modeling to obtain the filter.

Using sparse deconvolution, with the output signal $y(n)$ being the speech waveform, and with impulse response h obtained from AR modeling, we obtain the input signal $x(n)$, shown in Fig. 8. To do the sparse deconvolution, we minimized (6) using the program `deconvL1` in Sec. 7. The signal $x(n)$ is quite spiky and sparse. Note that the convolution $r(n) = (h * x)(n)$ is not equal to $y(n)$ exactly, because the cost function (6) does not enforce the equality. The cost function is intended for the case where the observed signal is noisy, which is not the case here. Therefore, the signal $r(n) = (h * x)(n)$, shown in Fig. 8, is only an approximate reconstruction of the original speech waveform. The parameter λ can be used to trade-off between reconstruction error and sparsity of $x(n)$.

The exact deconvolution of the signal $x(n)$ can be easily obtained because the filter $h(n)$ is a minimum-phase all-pole filter. The exact inverse system is therefore an FIR filter. Applying the FIR filter $1/H(z)$ to the speech waveform yields the exact deconvolution, shown in Fig. 8. When this signal is used as input to the filter $H(z)$, then the output will be exactly the speech waveform $x(n)$. However, as clear in the figure, the exact deconvolution yields a signal that is not sparse (compare with signal $x(n)$).

For the purpose of speech coding, the sparse input signal $x(n)$ has an advantage compared to the non-sparse signal: it is more readily compressed.

6 Conclusion

If it is known or expected that the unknown input signal x to an LTI system is sparse (e.g. 'spiky'), then sparse deconvolution is an appropriate approach for estimating x. It is assumed in these notes that the system is known and is described by a recursive (or non-recursive) difference equation, and that the available output signal is corrupted by additive white Gaussian noise.

Sparse deconvolution can be formulated as the minimization of a sparsity-regularized inverse problem. Although not the only choice, the ℓ_1 norm is a common choice of regularizer due to its being convex.

Sparse deconvolution based on ℓ_1 norm minimization can be implemented efficiently using:

1. A recursive difference equation as a model for the convolution system.

2. The majorization-minimization algorithm (with MM applied to the ℓ_1 norm).

3. Fast algorithms for the solution of banded systems of linear equations.

The regularization parameter λ can be set according to the noise variance (assuming the noise is stationary and its variance is known).

When *both* the filter and the input signal are unknown, then the problem is more difficult. This is the *blind* deconvolution problem. Sparsity and related non-Gaussian-based approaches can be effective for blind deconvolution; for example, see Refs. [7, 10, 18].

7 MATLAB program

```
function [x, cost] = deconvL1(y, lam, b, a, Nit)
% [x, cost] = deconvL1(y, lam, b, a, Nit)
% Sparse deconvolution by L1 norm minimization
% Cost function : 0.5 * sum(abs(y-H(x)).^2) + lam * sum(abs(x));
%
% INPUT
%   y - noisy data
%   b, a - filter coefficients for LTI convolution system H
%   lam - regularization parameter
%   Nit - number of iterations
%
% OUTPUT
%   x - deconvolved sparse signal
%   cost - cost function history

% Ivan Selesnick, selesi@poly.edu, 2012
% Algorithm: majorization-minimization with banded system solver.


y = y(:);                                               % convert column vector
cost = zeros(1, Nit);                                   % cost function history
N = length(y);

Nb = length(b);
Na = length(a);
B = spdiags(b(ones(N,1), :), 0:-1:1-Nb, N, N);          % create sparse matrices
A = spdiags(a(ones(N,1), :), 0:-1:1-Na, N, N);

H = @(x) A\(B*x);                                       % filter
AAT = A*A';                                             % A*A' : sparse matrix
x = y;                                                  % initialization
g = B'*(A'\y);                                          % H'*y

for k = 1:Nit
    W = (1/lam) * spdiags(abs(x), 0, N, N);                 % W : diag(abs(x)) (sparse)
    F = AAT + B*W*B';                                       % F : banded matrix (sparse)
    x = W * (g - (B'*(F\(B*(W*g)))));                       % update
    cost(k) = 0.5*sum(abs(y-H(x)).^2) + lam*sum(abs(x));    % cost function value
end
```

References

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK's user's guide*. SIAM, 1992. http://www.netlib.org/lapack.

[2] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

[3] R. G. Baraniuk, E. Candes, M. Elad, and Y. Ma, editors. Special issue on applications of sparse representation and compressive sensing. *Proc. IEEE*, 98(6), June 2010.

[4] M. S. O' Brien, A. N. Sinclair, and S. M. Kramer. Recovery of a sparse spike time series by L1 norm deconvolution. *IEEE Trans. Signal Process.*, 42(12):3353–3365, December 1994.

[5] S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998.

[6] J. F. Claerbout and F. Muir. Robust modeling of erratic data. *Geophysics*, 38(5):826–844, 1973.

[7] D. Donoho. On minimum entropy deconvolution. In *Applied Time Series Analysis II*, pages 565–609. Academic Press, 1981.

[8] M. Figueiredo, J. Bioucas-Dias, and R. Nowak. Majorization-minimization algorithms for wavelet-based image restoration. *IEEE Trans. Image Process.*, 16(12):2980–2991, December 2007.

[9] M. Figueiredo, J. Bioucas-Dias, J. P. Oliveira, and R. D. Nowak. On total-variation denoising: A new majorization-minimization algorithm and an experimental comparison with wavelet denoising. In *Proc. IEEE Int. Conf. Image Processing*, pages 2633–2636, 2006.

[10] D. Giacobello, M. G. Christensen, M. N. Murthi, S. H. Jensen, and M. Moonen. Sparse linear prediction and its applications to speech processing. *IEEE Trans. on Audio, Speech, and Lang. Proc.*, 20(5):1644–1657, July 2012.

[11] K. F. Kaaresen. Deconvolution of sparse spike trains by iterated window maximization. *IEEE Trans. Signal Process.*, 45(5):1173–1183, May 1997.

[12] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C: the art of scientific computing (2nd ed.)*. Cambridge University Press, 1992.

[13] B. D. Rao, K. Engan, S. F. Cotter, J. Palmer, and K. Kreutz-Delgado. Subset selection in noise based on diversity measure minimization. *IEEE Trans. Signal Process.*, 51(3):760–770, March 2003.

[14] I. Selesnick. Introduction to sparsity in signal processing. *Connexions*, 2012. http://cnx.org/content/m43545/1.3/.

[15] I. Selesnick. Total variation denoising (an MM algorithm). *Connexions*, 2012. http://cnx.org/content/m44934/1.2/.

[16] H. L. Taylor, S. C. Banks, and J. F. McCoy. Deconvolution with the l1 norm. *Geophysics*, 44(1):39–52, 1979.

[17] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc., Ser. B*, 58(1):267–288, 1996.

[18] R. A. Wiggins. Minimum entropy deconvolution. *Geoexploration*, 16(1-2):21–35, 1978.