Sparse Solutions of Systems of Equations and Sparse Modeling of Signals and Images: Midyear Report

Alfredo Nava-Tudela
ant@umd.edu
John J. Benedetto
Department of Mathematics
jjb@umd.edu

Abstract

We are interested in finding sparse solutions to systems of linear equations $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is underdetermined and fully-ranked, as presented in [1]. In this report we examine an implementation of the *orthogonal matching pursuit* algorithm, an algorithm to find sparse solutions to equations like the one described above, and present a logic for its validation and corresponding validation results.

This is the midyear report for the class project in AMSC 663/664. For a detailed description of the project, please see [4].

1 Introduction and context

Let n and m be two positive natural numbers such that $n < m$, and consider a full rank matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$. Given a column vector $\mathbf{b} \in \mathbb{R}^n$, we know that there is an infinite number of solutions to the system of linear equations

$$\mathbf{Ax} = \mathbf{b}, \tag{1}$$

where $\mathbf{x}$ is a column vector in $\mathbb{R}^m$ [5]. That is, there is an infinite number of column vectors $\mathbf{x} \in \mathbb{R}^m$ that satisfy equation (1). However, of this infinite number of possible solutions, we are interested in those solutions that are the *sparsest*. By this we mean solutions where $\mathbf{x}$ has the fewest number of non-zero entries. We shall make this concept more precise in section 2.

1.1 Why is this problem of interest?

Finding sparse solutions to systems of linear equations has many signal processing applications, among them, signal compression.

For example, the media encoding standard JPEG [2, 8] and its successor, JPEG-2000 [7], are both based on the notion of transform encoding. The JPEG standard uses the Discrete Cosine Transform (DCT), and the JPEG-2000 standard uses the Discrete Wavelet Transform (DWT). Both JPEG standards use properties of the DCT or the DWT, respectively, to achieve compression by creating approximations that represent the original image in a sparse way. We shall revisit this application in the second half of this project.

2 Defining the problem of finding sparse solutions to $\mathbf{Ax} = \mathbf{b}$

Consider a full-rank matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ with $n < m$, and define the underdetermined system of linear equations $\mathbf{Ax} = \mathbf{b}$. From the infinite number of solutions, we shall narrow down the choice to a well-defined solution by introducing a real valued function $\mathcal{J}(\mathbf{x})$ to evaluate the desirability of a would-be solution $\mathbf{x} \in \mathbb{R}^m$, with smaller values of $\mathcal{J}$ being preferred. This way, we can define the general optimization problem $(P_{\mathcal{J}})$ as

$$(P_{\mathcal{J}}): \qquad \min_{\mathbf{x}} \mathcal{J}(\mathbf{x}) \quad \text{subject to} \quad \mathbf{Ax} = \mathbf{b}. \tag{2}$$

Selecting a strictly convex function $\mathcal{J}(\cdot)$ guarantees a unique solution. For example if $\mathcal{J}(\mathbf{x}) = ||\mathbf{x}||_2^2$, the squared Euclidean norm of $\mathbf{x}$, the problem (P_2) that results from this choice has the unique minimum-norm solution $\hat{\mathbf{x}}$ given by

$$\hat{\mathbf{x}} = \mathbf{A}^+\mathbf{b} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{b}.$$

We know that the squared ℓ^2 norm is a measure of energy; we are interested in measures of *sparsity*. As was mentioned before, a vector $\mathbf{x}$ is sparse if there are few nonzero elements in the possible entries in $\mathbf{x}$. As such we shall introduce the ℓ^0 "norm"

$$||\mathbf{x}||_0 = \#\{i : x_i \neq 0\}.$$

Thus, if $||\mathbf{x}||_0 \ll n$, then $\mathbf{x}$ is sparse.

Consider the problem (P_0) obtained from the general prescription (2) that results from choosing $\mathcal{J}(\mathbf{x}) = ||\mathbf{x}||_0$, viz.,

$$(P_0): \qquad \min_{\mathbf{x}} ||\mathbf{x}||_0 \quad \text{subject to} \quad \mathbf{Ax} = \mathbf{b},$$

or its approximation (P_0^ϵ),

$$(P_0^\epsilon): \qquad \min_{\mathbf{x}} ||\mathbf{x}||_0 \quad \text{subject to} \quad ||\mathbf{Ax} - \mathbf{b}||_2 < \epsilon. \tag{3}$$

Unfortunately, the discrete and discontinuous nature of the ℓ^0 norm impedes the application of the standard convex analysis ideas that were at the core of the solution of (P_2). Moreover, it has been proven that finding a solution to (P_0^ϵ) is NP-hard [3], p. 228. However, the solution of (P_0^ϵ) can still be obtained by *greedy algorithms* when a sufficiently sparse solution exists. We introduce one such greedy algorithm next.

3 Orthogonal Matching Pursuit

In the first half of this project, we are interested in implementing and validating one of the many greedy algorithms (GAs) that attempt to solve (P_0). The general idea is as follows. Starting from $\mathbf{x}^0 = \mathbf{0}$, a greedy strategy iteratively constructs a k-term approximation $\mathbf{x}^k$ by maintaining a set of active columns—initially empty—and, at each stage, expanding that set by one additional column. The column chosen at each stage maximally reduces the residual ℓ^2 error in approximating $\mathbf{b}$ from the current set of active columns. After constructing an approximation including the new column, the residual error ℓ^2 is evaluated; if it now falls below a specified threshold, the algorithm terminates.

Orthogonal Matching Pursuit (OMP)—a GA for approximating the solution of (P_0):

Task: Approximate the solution of $(P_0) : \min_{\mathbf{x}} ||\mathbf{x}||_0$ subject to $\mathbf{Ax} = \mathbf{b}$.

Parameters: We are given the matrix $\mathbf{A}$, the vector $\mathbf{b}$, and the threshold ϵ_0.

Initialization: Initialize $k = 0$, and set

- The initial solution $\mathbf{x}^0 = \mathbf{0}$.
- The initial residual $\mathbf{r}^0 = \mathbf{b} - \mathbf{Ax}^0 = \mathbf{b}$.
- The initial solution support $\mathcal{S}^0 = Support\{\mathbf{x}^0\} = \emptyset$.

Main Iteration: Increment k by 1 and perform the following steps:

- **Sweep:** Compute the errors $\epsilon(j) = \min_{z_j} ||z_j\mathbf{a}_j - \mathbf{r}^{k-1}||_2^2$ for all j using the optimal choice $z_j^* = \mathbf{a}_j^T\mathbf{r}^{k-1}/||\mathbf{a}_j||_2^2$.
- **Update Support:** Find a minimizer j_0 of $\epsilon(j)$: $\forall j \notin \mathcal{S}^{k-1},\ \epsilon(j_0) \leq \epsilon(j)$, and update $\mathcal{S}^k = \mathcal{S}^{k-1} \cup \{j_0\}$.
- **Update Provisional Solution:** Compute $\mathbf{x}^k$, the minimizer of $||\mathbf{Ax} - \mathbf{b}||_2^2$ subject to $Support\{\mathbf{x}\} = \mathcal{S}^k$.
- **Update Residual:** Compute $\mathbf{r}^k = \mathbf{b} - \mathbf{Ax}^k$.
- **Stopping Rule:** If $||\mathbf{r}^k||_2 < \epsilon_0$, stop. Otherwise, apply another iteration.

Output: The proposed solution is $\mathbf{x}^k$ obtained after k iterations.

This algorithm is known in the literature of signal processing by the name *orthogonal matching pursuit* (OMP), and this is the algorithm we have implemented and validated. OMP solves, in essence, (P_0^ϵ) for $\epsilon = \epsilon_0$, a given positive threshold. See (3) above for details.

4 An OMP Implementation

For a given matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$, if the approximation delivered by OMP has k_0 zeros, the method requires $\mathcal{O}(k_0mn)$ flops in general; this can be dramatically better than the exhaustive search, which requires $\mathcal{O}(nm^{k_0}k_0^2)$ flops.

We would like to make the following observations about the OMP algorithm described in section 3. The step that updates the provisional solution seeks to minimize $||\mathbf{Ax} - \mathbf{b}||_2^2$,

subject to $Support\{\mathbf{x}\} = \mathcal{S}^k$. This is equivalent to solving the least squares approximation problem $\min_{\tilde{\mathbf{x}}} ||\mathbf{A}^{(k)}\tilde{\mathbf{x}} - \mathbf{b}||_2^2$ for the matrix $\mathbf{A}^{(k)}$ that results from using only the k active columns of $\mathbf{A}$ defined by $\mathcal{S}^k$, and $\tilde{\mathbf{x}}$ is the vector in $\mathbb{R}^k$ whose i-th entry corresponds to the column of $\mathbf{A}$ that was chosen during the i-th iteration of the main loop. See figure 1.

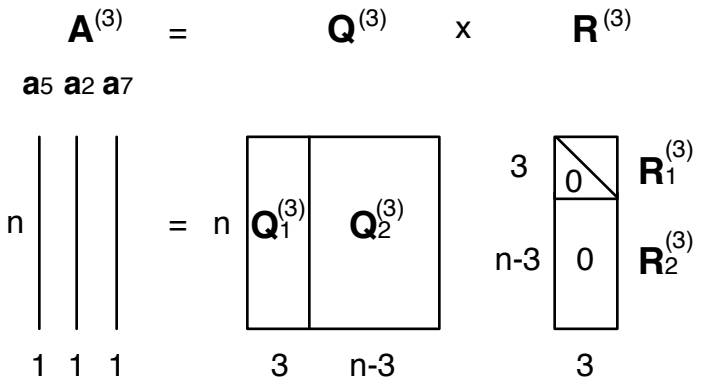

Figure 1: Suppose that after $k = 3$ iterations of the main loop, OMP has chosen, in the following order, columns $\mathbf{a}_5$, $\mathbf{a}_2$, and $\mathbf{a}_7$ from matrix $\mathbf{A}$. We form sub-matrix $\mathbf{A}^{(3)} = (\mathbf{a}_5\ \mathbf{a}_2\ \mathbf{a}_7)$, and its QR decomposition $\mathbf{A}^{(3)} = \mathbf{Q}^{(3)}\mathbf{R}^{(3)}$, which we use to solve the least-squares problem $||\mathbf{A}^{(3)}\tilde{\mathbf{x}} - \mathbf{b}||_2^2 = 0$, with $\tilde{\mathbf{x}} \in \mathbb{R}^3$.

For the case when $\mathbf{A}$ is a relatively small matrix, we can solve this problem, for example, by factorizing $\mathbf{A}^{(k)} = \mathbf{Q}^{(k)}\mathbf{R}^{(k)}$ with the QR-algorithm, and then observing that

$$\mathbf{A}^{(k)} = \mathbf{Q}^{(k)}\mathbf{R}^{(k)} = \mathbf{Q}_1^{(k)}\mathbf{R}_1^{(k)} + \mathbf{Q}_2^{(k)}\mathbf{R}_2^{(k)} = \mathbf{Q}_1^{(k)}\mathbf{R}_1^{(k)} + \mathbf{0} = \mathbf{Q}_1^{(k)}\mathbf{R}_1^{(k)}, \tag{4}$$

where—using Matlab notation—

$$\mathbf{Q}_1^{(k)} = \mathbf{Q}^{(k)}(:, 1{:}k),\ \mathbf{Q}_2^{(k)} = \mathbf{Q}^{(k)}(:, k{+}1{:}n),\ \mathbf{R}_1^{(k)} = \mathbf{R}^{(k)}(1{:}k, :),\ \text{and}\ \mathbf{R}_2^{(k)} = \mathbf{R}^{(k)}(k{+}1{:}n, :).$$

Then, from equation 4, we have

$$\begin{aligned}
\mathbf{A}^{(k)}\tilde{\mathbf{x}}_0 = \mathbf{b} &\Leftrightarrow \mathbf{Q}_1^{(k)}\mathbf{R}_1^{(k)}\tilde{\mathbf{x}}_0 = \mathbf{b} \\
&\Rightarrow \mathbf{Q}_1^{(k)\mathrm{T}}\mathbf{Q}_1^{(k)}\mathbf{R}_1^{(k)}\tilde{\mathbf{x}}_0 = \mathbf{Q}_1^{(k)\mathrm{T}}\mathbf{b} \\
&\Leftrightarrow \mathbf{R}_1^{(k)}\tilde{\mathbf{x}}_0 = \mathbf{Q}_1^{(k)\mathrm{T}}\mathbf{b} \\
&\Leftrightarrow \tilde{\mathbf{x}}_0 = (\mathbf{R}_1^{(k)})^{-1}\mathbf{Q}_1^{(k)\mathrm{T}}\mathbf{b},
\end{aligned}$$

where $\tilde{\mathbf{x}}_0 \in \mathbb{R}^k$ is the solution to the equivalent minimization problem described above. Finally, when OMP returns successfully after k_0 iterations, we embed $\tilde{\mathbf{x}}_0 \in \mathbb{R}^{k_0}$ in $\mathbf{0} \in \mathbb{R}^m$ "naturally" to obtain the solution $\mathbf{x}_0 \in \mathbb{R}^m$ to the initial least-squares approximation problem $||\mathbf{A}\mathbf{x}-\mathbf{b}||_2^2$ subject to the final active column set $\mathcal{S}^{k_0}$. The natural embedding refers to setting the j-th entry of $\mathbf{0} \in \mathbb{R}^m$ equal to the i-th entry in $\tilde{\mathbf{x}}_0 \in \mathbb{R}^{k_0}$ if during the i-th loop of the main algorithm, OMP chose the j-th column of $\mathbf{A}$.

5 OMP Validation Protocol and Validation Results

In this section we present the validation protocol that we followed to verify the correctness of our OMP implementation.

5.1 Theoretical results that motivate and justify the protocol

The following results provide the foundation for the validation protocol that we adopted. This protocol can be used to validate any OMP implementation.

Given a matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ with $n < m$, we can compute its mutual coherence defined as follows.

Definition 1. *The mutual coherence of a given matrix* $\mathbf{A}$ *is the largest absolute normalized inner product between different columns from* $\mathbf{A}$*. Denoting the k-th column in* $\mathbf{A}$ *by* $\mathbf{a}_k$*, the mutual coherence is given by*

$$\mu(\mathbf{A}) = \max_{1 \leq k,j \leq m,\ k \neq j} \frac{|\mathbf{a}_k^T \mathbf{a}_j|}{||\mathbf{a}_k||_2 \cdot ||\mathbf{a}_j||_2}. \tag{5}$$

The mutual coherence gives us a simple criterion by which we can test when a solution to (1) is the unique sparsest solution available. In what follows, we assume that $\mathbf{A} \in \mathbb{R}^{n \times m}$, $n < m$, and $\text{rank}(\mathbf{A}) = n$.

Lemma 1. *If* $\mathbf{x}$ *solves* $\mathbf{Ax} = \mathbf{b}$*, and* $||\mathbf{x}||_0 < \frac{1}{2}(1 + 1/\mu(\mathbf{A}))$*, then* $\mathbf{x}$ *is the sparsest solution. That is, if* $\mathbf{y} \neq \mathbf{x}$ *also solves the equation, then* $||\mathbf{x}||_0 < ||\mathbf{y}||_0$*.*

This same criterion can be used to test when OMP will find the sparsest solution.

Lemma 2. *For a system of linear equations* $\mathbf{Ax} = \mathbf{b}$*, if a solution* $\mathbf{x}$ *exists obeying* $||\mathbf{x}||_0 < \frac{1}{2}(1 + 1/\mu(\mathbf{A}))$*, then an OMP run with threshold parameter* $\epsilon_0 = 0$ *is guaranteed to find* $\mathbf{x}$ *exactly.*

The proofs of these lemmas can be found or are inspired by results in [1]. In light of these lemmas, we can envision the following roadmap to validate an implementation of OMP. We have a simple unified theoretical criterion to guarantee both solution uniqueness and OMP convergence. The following theorem simply unifies the previous lemmas into one statement.

Theorem 3. *If* $\mathbf{x}$ *is a solution to* $\mathbf{Ax} = \mathbf{b}$*, and* $||\mathbf{x}||_0 < \frac{1}{2}(1 + \mu(\mathbf{A}))$*, then* $\mathbf{x}$ *is the unique sparsest solution to* $\mathbf{Ax} = \mathbf{b}$*, and OMP will find it.*

In light of this result, we can establish the following protocol to validate any implementation of OMP.

5.2 Protocol

Given a full-rank matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$, with $n < m$, compute $\mu(\mathbf{A})$, and find the largest integer k smaller than or equal to $\frac{1}{2}(1 + 1/\mu(\mathbf{A}))$. That is, $k = \lfloor \frac{1}{2}(1 + 1/\mu(\mathbf{A})) \rfloor$.

Then, build a vector $\mathbf{x}$ with exactly k non-zero entries and produce a right hand side vector $\mathbf{b} = \mathbf{A}\mathbf{x}$. This way, you have a known sparsest solution $\mathbf{x}$ to which to compare the output of any OMP implementation.

Pass $\mathbf{A}$, $\mathbf{b}$, and ϵ_0 to OMP to produce a solution vector $\mathbf{x}_{OMP} = \text{OMP}(\mathbf{A}, \mathbf{b}, \epsilon_0)$.

If OMP terminates after k iterations (or less), and $||\mathbf{A}\mathbf{x}_{OMP} - \mathbf{b}|| < \epsilon_0$, for all possible $\mathbf{x}$ and $\epsilon_0 > 0$, then the OMP implementation would have been validated.

5.3 Results

Call $\kappa_{\mathbf{A}} = \frac{1}{2}(1 + 1/\mu(\mathbf{A}))$, the constant dependent on $\mathbf{A}$ that guarantees the results of Theorem 3 for matrix $\mathbf{A}$. To test our implementation, we ran two experiments involving two random matrices.

1. $\mathbf{A}_1 \in \mathbb{R}^{100\times 200}$, with entries in the Gaussian distribution $N(0,1)$, i.i.d., for which its mutual coherence turned out to be $\mu(\mathbf{A}_1) = 0.3713$, corresponding to $k = 1 = \lfloor \kappa_{\mathbf{A}_1} \rfloor$.

2. $\mathbf{A}_2 \in \mathbb{R}^{200\times 400}$, with entries in the Gaussian distribution $N(0,1)$, i.i.d., for which its mutual coherence turned out to be $\mu(\mathbf{A}_2) = 0.3064$, corresponding to $k = 2 = \lfloor \kappa_{\mathbf{A}_2} \rfloor$.

We first note that, with probability 1, $\mathbf{A}_i$, $(i = 1, 2)$, is a full-rank matrix [1]. Second, we would like to mention that for full-rank matrices $\mathbf{A}$ of size $n \times m$, the mutual coherence satisfies $\mu(\mathbf{A}) \geq \sqrt{(m-n)/(n \cdot (m-1))}$, with the equality being sharp [6]. We used these results to guide us into obtaining matrix $\mathbf{A}_2$ for which $k = 2 = \lfloor \kappa_{\mathbf{A}_2} \rfloor > 1$.

For each matrix $\mathbf{A}_i$, $(i = 1, 2)$, we chose 100 compatible vectors with k non-zero entries whose positions were chosen at random, and whose entries were in the Gaussian distribution $N(0,1)$, i.i.d..

Then, for each such vector $\mathbf{x}$, we built a corresponding right hand side vector $\mathbf{b} = \mathbf{A}_i\mathbf{x}$. Each of these vectors would then be the unique sparsest solution to $\mathbf{A}_i\mathbf{x} = \mathbf{b}$, and OMP should be able to find them.

Finally, given $\epsilon_0 > 0$, if our implementation of OMP were correct, it should stop after k steps (or less), and if $\mathbf{x}_{OMP} = \text{OMP}(\mathbf{A}_i, \mathbf{b}, \epsilon_0)$, then $||\mathbf{b} - \mathbf{A}_i\mathbf{x}_{OMP}|| < \epsilon_0$.

We ran these experiments for twelve values of ϵ_0 equal to 10, 1, 10^{-1}, 10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}, 10^{-12}, 10^{-14}, 10^{-15}, and 10^{-16}. For each of these values of ϵ_0 we built 100 vectors as described above, with their respective right hand side vectors, both of which were fed to OMP together with the tolerance ϵ_0 being tested.

We kept track of how many iterations it took OMP to stop, and the value of the norm of the residue $||\mathbf{b} - \mathbf{A}_i\mathbf{x}_{OMP}||$ at the end of each run. We mention that our implementation of OMP had as stopping condition that either the residue would be less the tolerance ϵ_0 given, or that n iterations of the main loop would have been executed.

Figure 2 shows the summary of the results for matrix $\mathbf{A}_1$. It contains two graphs, the top graph represents the average of the norm of the residue over the 100 experiments executed for a given tolerance, versus the 12 tolerances chosen. The red line represents the identity in this case. The second graph is the same but for the average number of iterations it took OMP to stop vs the tolerances chosen. The red line in this case is the expected number of iterations $k = \lfloor \kappa_{\mathbf{A}_1} \rfloor$ at stop time. Figure 4 is the same as figure 2, but for matrix $\mathbf{A}_2$.

One can observe in both cases that there are three modal behaviors of OMP. The rightmost points in each graph correspond to tolerances ϵ_0 that are "too large". For them, OMP converges, but it does not have to do much work necessarily, since the default initial solution $\mathbf{x} = \mathbf{0}$ is already close to the right hand side $\mathbf{b}$. The typical behavior corresponds to points in the middle of the graph, they represent the cases when OMP converges in exactly k iterations to the sparsest solution within machine precision. And, finally, the leftmost points, they represent when OMP fails to converge because the tolerances ϵ_0 are too close to machine precision, basically trampling OMP efforts to converge due to roundoff and truncation errors.

In figure 3 we exemplified each of the three modal behaviors with three values of ϵ_0 typical of each mode. The figure contains three graphs, the top graph is for $\epsilon_0 = 10$, the middle graph is for $\epsilon_0 = 10^{-6}$, and the bottom graph is for $\epsilon_0 = 10^{-16}$. Each of the graphs shows the individual results for each of the 100 experiments ran for each tolerance ϵ_0. This figure corresponds to matrix $\mathbf{A}_1$. In figure 5 we have the same graphs but for matrix $\mathbf{A}_2$.

6 Conclusion

Our validation protocol and results confirm that our implementation of OMP is correct. This implementation will return a solution $\mathbf{x} = \text{OMP}(\mathbf{A}, \mathbf{b}, \epsilon_0)$ to $\mathbf{Ax} = \mathbf{b}$, within machine precision, whenever the tolerance $\epsilon_0 \geq 10^{-14}$, and provided $||\mathbf{x}||_0 \leq \kappa_{\mathbf{A}}$.

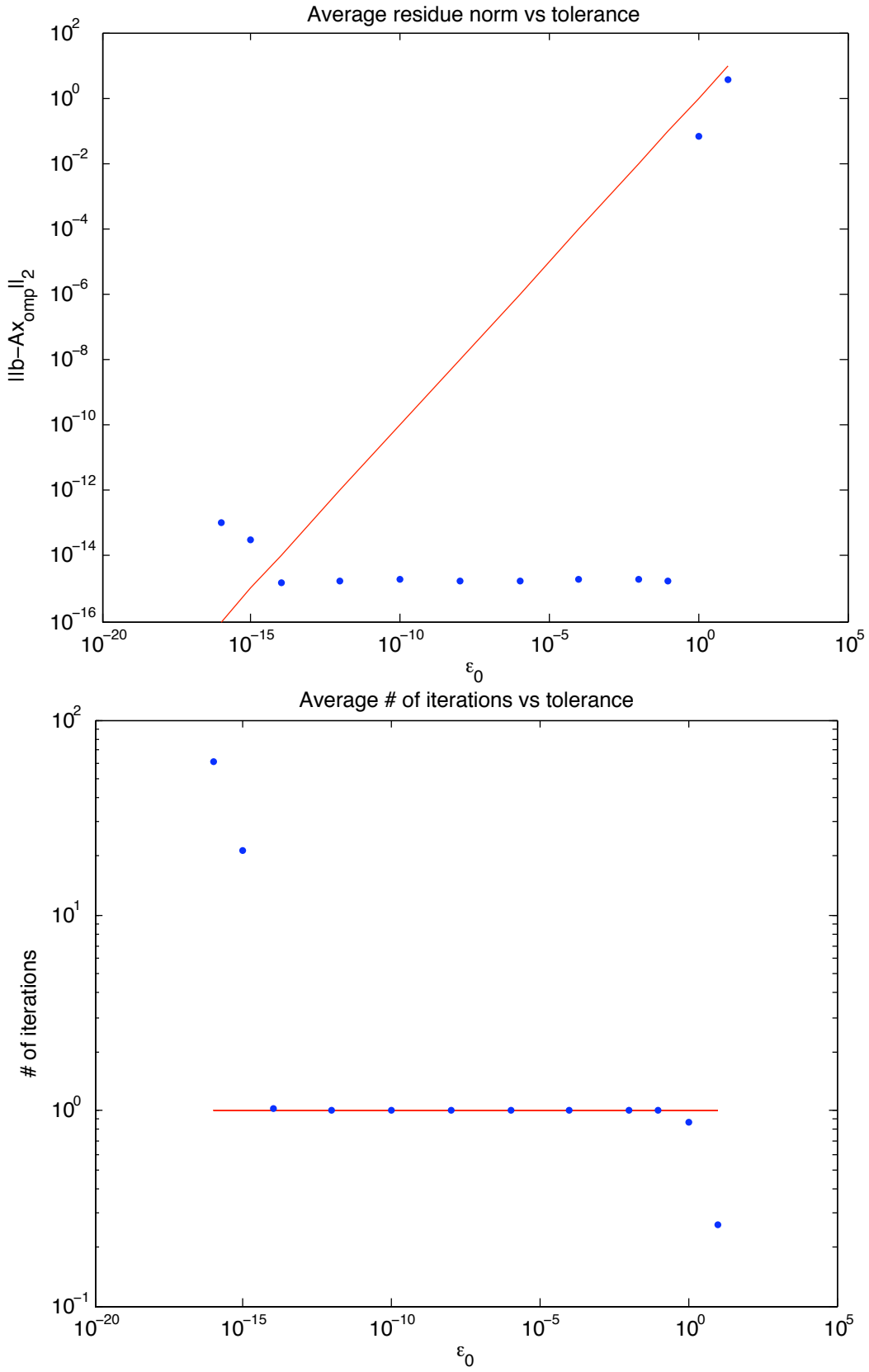

Figure 2: OMP behavior for a matrix $\mathbf{A}$ with $\mu(\mathbf{A}) = 0.3713$, which corresponds to $k = 1$.

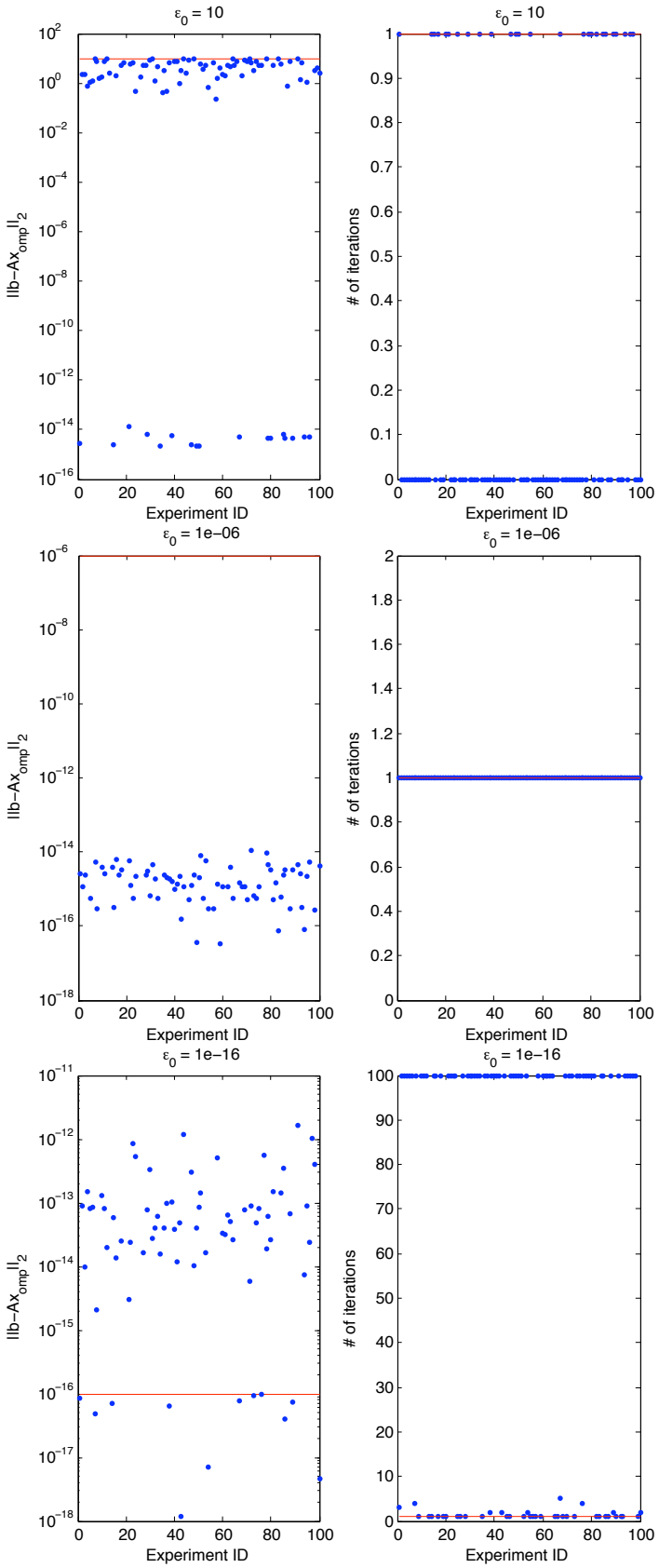

Figure 3: The three modal behaviors, dependent on ϵ_0, observed for the matrix **A** used in Fig. 2.

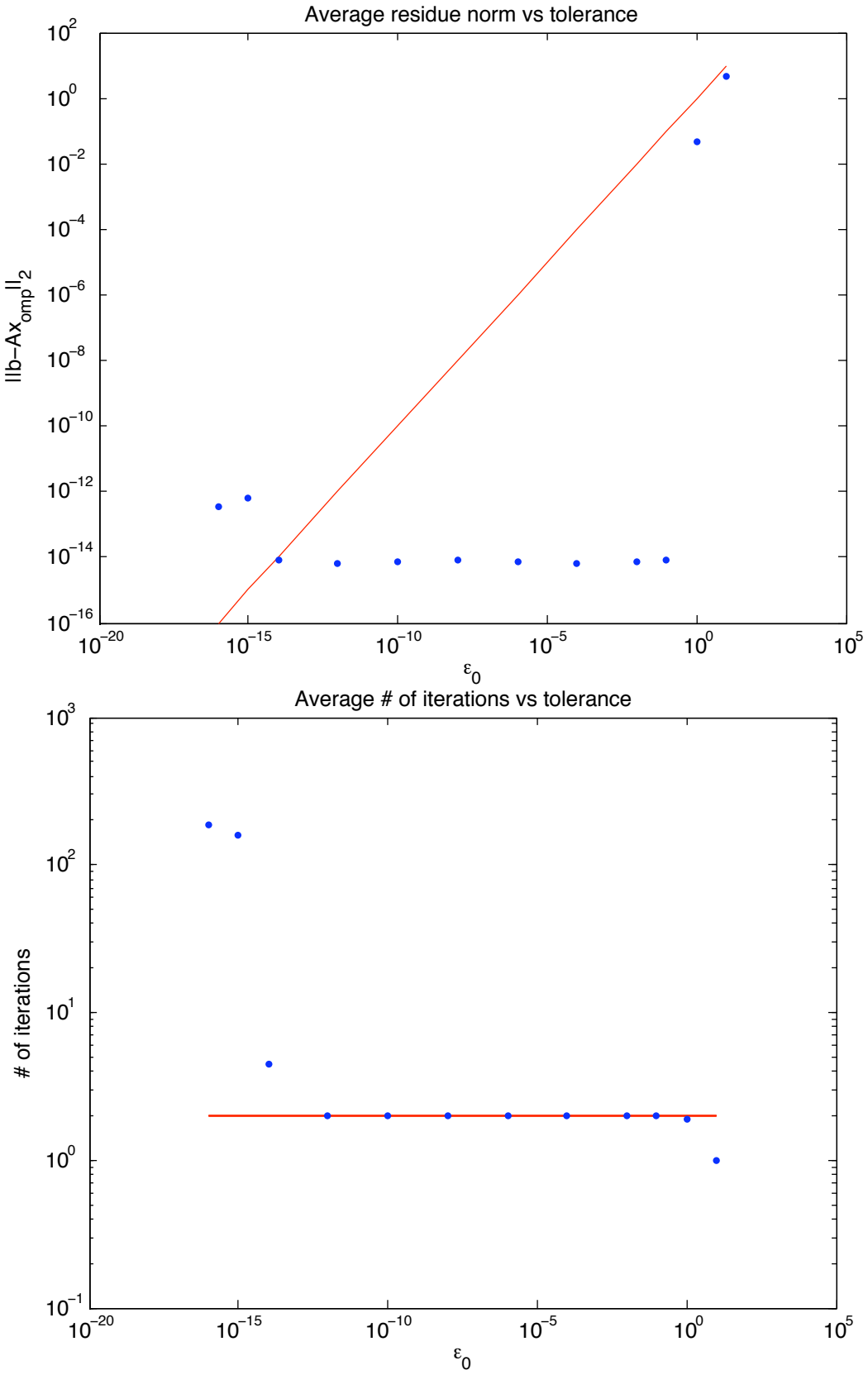

Figure 4: OMP behavior for a matrix $\mathbf{A}$ with $\mu(\mathbf{A}) = 0.3064$, which corresponds to $k = 2$.

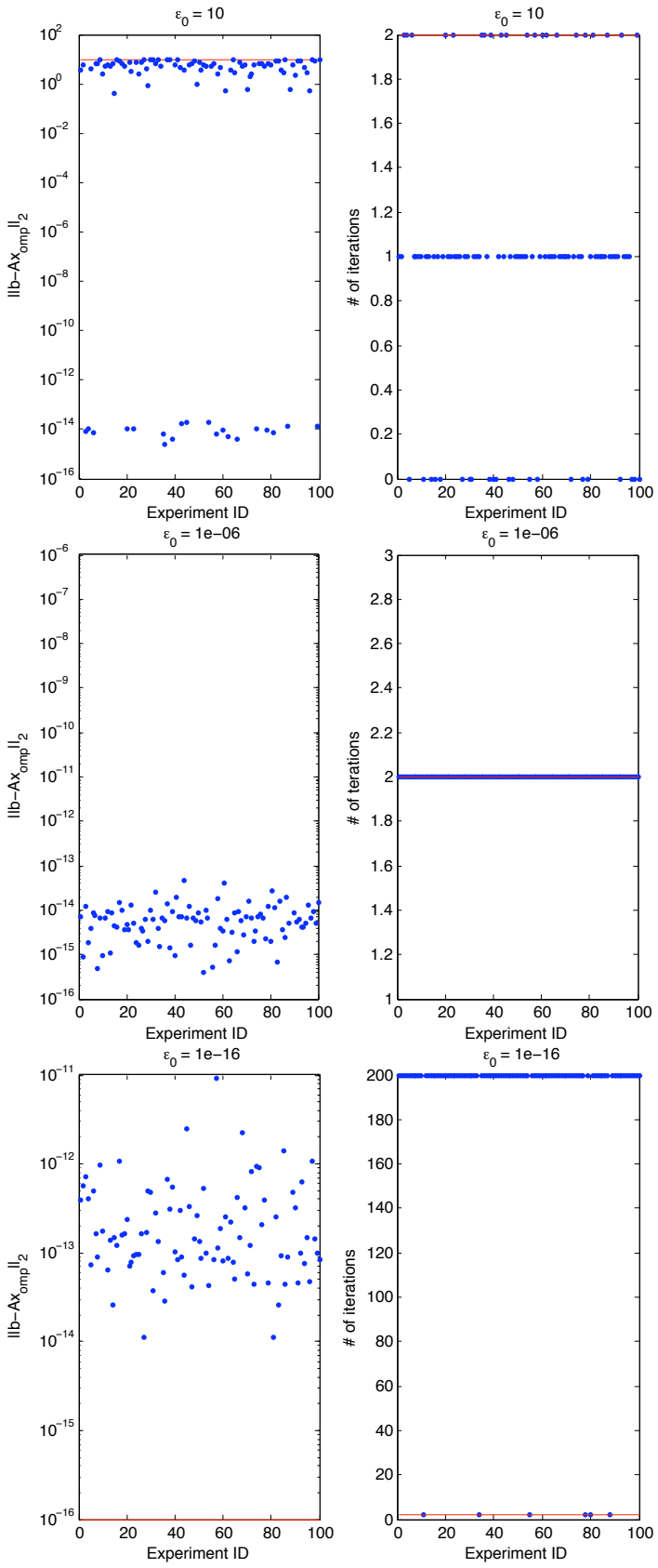

Figure 5: The three modal behaviors, dependent on ϵ_0, observed for the matrix **A** used in Fig. 4.

References

[1] A. M. Bruckstein, D. L. Donoho, and M. Elad, *From sparse solutions of systems of equations to sparse modeling of signals and images*, SIAM Review, 51 (2009), pp. 34–81.

[2] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1998.

[3] B. K. Natarajan, *Sparse approximate solutions to linear systems*, SIAM Journal on Computing, 24 (1995), pp. 227–234.

[4] A. Nava-Tudela, *Sparse solutions of systems of equations and sparse modeling of signals and images*. AMSC 663/664 Class Project Proposal, 2010.

[5] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, 1973.

[6] T. Strohmer and R. W. Heath, *Grassmanian frames with applications to coding and communication*, Appl. Comput. Harmon. Anal., 14 (2004), pp. 257–275.

[7] D. S. Taubman and M. W. Mercellin, *JPEG 2000: Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, 2001.

[8] G. K. Wallace, *The JPEG still picture compression standard*, Communications of the ACM, 34 (1991), pp. 30–44.