EECS 16A Designing Information Devices and Systems I
Spring 2019 Lecture Notes Note 25

25.1 Speeding up OMP

In the last lecture note, we introduced orthogonal matching pursuit (OMP), an algorithm that can extract information from sparse signals. Recall the setup: we have n devices that could potentially be sending a signal, but at most k are on. The signal $\vec{y}$ we receive is the sum of the signal from each device.

Specifically, let $A \in \mathbb{R}^{m \times n}$ be a matrix with $n > m$ (short and fat), where the jth column represents the song sent by device j:

$$A = \begin{bmatrix} | & | & & | \\ \vec{S}_1 & \vec{S}_2 & \dots & \vec{S}_n \\ | & | & & | \end{bmatrix}$$

Then, let $\vec{x} \in \mathbb{R}^n$ be a vector where the jth entry represents the "volume" of the jth device. Our received signal is

$$A\vec{x} = \begin{bmatrix} | & | & & | \\ \vec{S}_1 & \vec{S}_2 & \dots & \vec{S}_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \sum_{j=1}^{n} x_j \vec{S}_j = \vec{y}.$$

From $\vec{y}$, we want to find $\vec{x}$, given that at most $k < m$ entries of $\vec{x}$ are non-zero. OMP gives us a way to find which k entries are "on." Then, we can form a matrix A_k with just the columns for the "on" devices, and we can find $\vec{x}$ by inverting A_k.

At a high level, in each iteration of the algorithm, we find one "on" device by correlating the signal $\vec{y}$ with each column of A and taking the maximum. Then, we find the residual, or the part of $\vec{y}$ still unexplained by our current list of "on" devices. Using this residual, we repeat to find one more "on" device to add to our list. Eventually, we find all k "on" devices, allowing us to solve for $\vec{x}$.

Let A_j denote the list of j "on" devices at iteration j. To find the residual $\vec{r}$, we need to project $\vec{y}$ onto the columns of A_j and then subtract this projection from $\vec{y}$ (recall the projection formula from the least squares unit):

$$\vec{r} = \vec{y} - A_j(A_j^T A_j)^{-1} A_j^T \vec{y}.$$

However, matrix inversion is computationally expensive - for an $n \times n$ matrix, inversion is $O(n^3)$. We have to do inversion at every time step, making our algorithm slow.

Is there a way to avoid doing such computations? Yes! All we are doing is projecting $\vec{y}$ onto the j vectors $\{\vec{S}_{i_1}, \vec{S}_{i_2}, \dots, \vec{S}_{i_j}\}$. For general $\vec{S}_i$, we need to compute the projection matrix $A_j(A_j^T A_j)^{-1} A_j^T$ using least squares. But if the $\vec{S}_i$ are orthogonal, meaning that $\vec{S}_i^T \vec{S}_j = 0$ for $i \neq j$, then we have another faster way of computing the projection.

25.2 Projection onto Orthogonal Vectors

In this section, we will show that if the columns of A_j are mutually orthogonal to each other, the projection of $\vec{y}$ onto span(A_j) is the sum of the projection of $\vec{y}$ onto each column of A_j. Recall that the projection of a vector $\vec{y}$ on to any other nonzero vector $\vec{b}$ of the same size is

$$\vec{y}_b = \frac{\vec{y}^T\vec{b}}{\left\|\vec{b}\right\|^2}\vec{b}. \tag{1}$$

Let's take a look at the case where $j = 2$ and the signatures are mutually orthogonal. Suppose the songs found so far are $\vec{S}_1$ and $\vec{S}_2$, i.e., $A_2 = \begin{bmatrix} | & | \\ \vec{S}_1 & \vec{S}_2 \\ | & | \end{bmatrix}$. Then, the projection of $\vec{y}$ onto A_2 is

$$\vec{y}_{A_2} = A_2\left(A_2^T A_2\right)^{-1} A_2^T \vec{y}. \tag{2}$$

Let's first compute the term $\left(A_2^T A_2\right)^{-1}$:

$$A_2^T A_2 = \begin{bmatrix} — & \vec{S}_1{}^T & — \\ — & \vec{S}_2{}^T & — \end{bmatrix} \begin{bmatrix} | & | \\ \vec{S}_1 & \vec{S}_2 \\ | & | \end{bmatrix} \tag{3}$$

$$= \begin{bmatrix} \vec{S}_1{}^T\vec{S}_1 & \vec{S}_1{}^T\vec{S}_2 \\ \vec{S}_2{}^T\vec{S}_1 & \vec{S}_2{}^T\vec{S}_2 \end{bmatrix} \tag{4}$$

$$= \begin{bmatrix} \left\|\vec{S}_1\right\|^2 & 0 \\ 0 & \left\|\vec{S}_2\right\|^2 \end{bmatrix}. \tag{5}$$

$$\tag{6}$$

Thus, we have

$$\left(A_2^T A_2\right)^{-1} = \begin{bmatrix} \frac{1}{\|\vec{S}_1\|^2} & 0 \\ 0 & \frac{1}{\|\vec{S}_2\|^2} \end{bmatrix}. \tag{7}$$

Then, substituting this matrix into the original expression, the projection of $\vec{y}$ onto span(A_2) is

$$\vec{y}_{A_2} = A_2\left(A_2^T A_2\right)^{-1} A_2^T \vec{y} \tag{8}$$

$$= \begin{bmatrix} | & | \\ \vec{S}_1 & \vec{S}_2 \\ | & | \end{bmatrix} \begin{bmatrix} \frac{1}{\|\vec{S}_1\|^2} & 0 \\ 0 & \frac{1}{\|\vec{S}_2\|^2} \end{bmatrix} \begin{bmatrix} — & \vec{S}_1^{\,T} & — \\ — & \vec{S}_2^{\,T} & — \end{bmatrix} \vec{y} \tag{9}$$

$$= \begin{bmatrix} | & | \\ \vec{S}_1 & \vec{S}_2 \\ | & | \end{bmatrix} \begin{bmatrix} \frac{1}{\|\vec{S}_1\|^2} & 0 \\ 0 & \frac{1}{\|\vec{S}_2\|^2} \end{bmatrix} \begin{bmatrix} \vec{S}_1^{\,T}\vec{y} \\ \vec{S}_2^{\,T}\vec{y} \end{bmatrix} \tag{10}$$

$$= \begin{bmatrix} | & | \\ \vec{S}_1 & \vec{S}_2 \\ | & | \end{bmatrix} \begin{bmatrix} \frac{\vec{S}_1^{\,T}\vec{y}}{\|\vec{S}_1\|^2} \\ \frac{\vec{S}_2^{\,T}\vec{y}}{\|\vec{S}_2\|^2} \end{bmatrix} \tag{11}$$

$$= \left(\frac{\vec{S}_1^{\,T}\vec{y}}{\left\|\vec{S}_1\right\|^2}\right)\vec{S}_1 + \left(\frac{\vec{S}_2^{\,T}\vec{y}}{\left\|\vec{S}_2\right\|^2}\right)\vec{S}_2. \tag{12}$$

Observe that the first term in the sum above is the projection of $\vec{y}$ onto $\vec{S}_1$ and the second term is the projection of $\vec{y}$ onto $\vec{S}_2$. Generalizing, the projection of $\vec{y}$ onto span(A_n) where A_n has mutually orthogonal columns is

$$\vec{y}_{A_n} = \left(\frac{\vec{S}_1^{\,T}\vec{y}}{\left\|\vec{S}_1\right\|^2}\right)\vec{S}_1 + \left(\frac{\vec{S}_2^{\,T}\vec{y}}{\left\|\vec{S}_2\right\|^2}\right)\vec{S}_2 + \cdots + \left(\frac{\vec{S}_n^{\,T}\vec{y}}{\left\|\vec{S}_n\right\|^2}\right)\vec{S}_n. \tag{13}$$

Furthermore, observe that if $\vec{S}_1,\ldots,\vec{S}_n$ are unit vectors (i.e., they all have length 1), then the above further reduces to

$$\vec{y}_{A_n} = \left(\vec{S}_1^{\,T}\vec{y}\right)\vec{S}_1 + \left(\vec{S}_2^{\,T}\vec{y}\right)\vec{S}_2 + \cdots + \left(\vec{S}_n^{\,T}\vec{y}\right)\vec{S}_n. \tag{14}$$

This observation that projection is faster with orthogonal vectors will motivate our use of Gram-Schmidt to speed up OMP.

25.3 Gram-Schmidt Process

Before we begin, let's remind ourselves that the following subspaces are equivalent for any pairs of linearly independent vectors $\vec{v}_1, \vec{v}_2$:

- span($\vec{v}_1$, $\vec{v}_2$)
- span($\vec{v}_1$, $\alpha\vec{v}_2$)
- span($\vec{v}_1$, $\vec{v}_1 + \vec{v}_2$)
- span($\vec{v}_1$, $\vec{v}_1 - \vec{v}_2$)
- span($\vec{v}_1$, $\vec{v}_2 - \alpha\vec{v}_1$)

First, as a preliminary, let's show that the projection error

$$\vec{q} = \vec{y} - \vec{y}_{\vec{b}} = \vec{y} - \frac{\vec{y}^T\vec{b}}{\left\|\vec{b}\right\|^2}\vec{b}$$

is always orthogonal to $\vec{b}$. To see this, what should α be if we would like $\vec{b}$ and $\vec{y} - \alpha\vec{b}$ to be orthogonal to each other? Let's solve this algebraically using the definition of orthogonality:

$$\vec{b}^T\left(\vec{y} - \alpha\vec{b}\right) = 0 \tag{15}$$

$$\vec{b}^T\vec{y} - \alpha\left\|\vec{b}\right\|^2 = 0 \tag{16}$$

$$\alpha = \frac{\vec{y}^T\vec{b}}{\left\|\vec{b}\right\|^2}. \tag{17}$$

We will use this fact later in this section.

Definition 25.1 (Orthonormal): A set of vectors $\{\vec{e}_1, \ldots, \vec{e}_n\}$ is **orthonormal** if all the vectors are mutually orthogonal to each other and all are of unit length.

Gram Schmidt is an algorithm that takes a set of linearly independent vectors $\{\vec{S}_1, \ldots, \vec{S}_n\}$ and generates an orthonormal set of vectors $\{\vec{e}_1, \ldots, \vec{e}_n\}$ that span the same vector space as the original set. Concretely, $\{\vec{e}_1, \ldots, \vec{e}_n\}$ needs to satisfy the following:

- span($\{S_1, \ldots, S_n\}$) = span($\{\vec{e}_1, \ldots, \vec{e}_n\}$)
- $\{\vec{e}_1, \ldots, \vec{e}_n\}$ is an orthonormal set of vectors.

Now let's see how we can do this with a set of three vectors $\{\vec{S}_1, \vec{S}_2, \vec{S}_3\}$ that is linearly independent of each other.

- **Step 1:** Find unit vector $\vec{e}_1$ such that span($\{\vec{e}_1\}$) = span($\{\vec{S}_1\}$).
  Since span($\{\vec{S}_1\}$) is a one dimensional vector space, we can simply scale $\{\vec{S}_1\}$ so that it is unit norm:

  $$\vec{e}_1 = \frac{\vec{S}_1}{\left\|\vec{S}_1\right\|}. \tag{18}$$

- **Step 2:** Given $\vec{e}_1$ from the previous step, find $\vec{e}_2$ such that $span(\{\vec{e}_1, \vec{e}_2\}) = span(\{\vec{S}_1, \vec{S}_2\})$ and orthogonal to $\vec{e}_1$. We know that $\vec{S}_2 -$ (the projection of $\vec{S}_2$ on $\vec{e}_1$) would be orthogonal to $\vec{e}_1$ as seen earlier. So first, we can find

  $$\vec{q}_2 = \vec{S}_2 - \left(\vec{S}_2^T\vec{e}_1\right)\vec{e}_1, \tag{19}$$

  which is orthogonal to $\vec{e}_1$. Then, we can normalize to get $\vec{e}_2 = \frac{\vec{q}_2}{\|\vec{q}_2\|}$. Note that these operations preserve the span because $\vec{e}_1$ and $\vec{e}_2$ are just linear combinations of $\vec{S}_1$ and $\vec{S}_2$.

- **Step 3:** Now given $\vec{e}_1$ and $\vec{e}_2$ in the previous steps, we would like to find $\vec{e}_3$ such that $span(\{\vec{e}_1,\vec{e}_2,\vec{e}_3\}) = span(\{\vec{S}_1,\vec{S}_2,\vec{S}_3\})$. We know that the projection of $\vec{S}_3$ onto the subspace spanned by $\vec{e}_1,\vec{e}_2$ is

$$\left(\vec{S}_3^T\vec{e}_2\right)\vec{e}_2+\left(\vec{S}_3^T\vec{e}_1\right)\vec{e}_1. \tag{20}$$

We know that

$$\vec{q}_3=\vec{S}_3-\left(\vec{S}_3^T\vec{e}_2\right)\vec{e}_2-\left(\vec{S}_3^T\vec{e}_1\right)\vec{e}_1 \tag{21}$$

is orthogonal to $\vec{e}_1$ and $\vec{e}_2$. Normalizing, we have $\vec{e}_3=\frac{\vec{q}_3}{\|\vec{q}_3\|}$.

We can generalize the above procedure for any number of linearly independent vectors as follows:

Inputs

- A set of linearly independent vectors $\{\vec{S}_1,\ldots,\vec{S}_n\}$.

Outputs

- An orthonormal set of vectors $\{\vec{e}_1,\ldots,\vec{e}_n\}$, where span($\{\vec{S}_1,\ldots,\vec{S}_n\}$) = span($\{\vec{e}_1,\ldots,\vec{e}_n\}$).

Gram Schmidt Procedure

- compute $\vec{e}_1$: $\vec{e}_1=\frac{\vec{S}_1}{\|\vec{S}_1\|}$
- for $(i=2\ldots n)$:
  1. Compute the vector $\vec{q}_i$, such that $span(\{\vec{e}_1,\ldots,\vec{q}_i\}) = span(\{\vec{S}_1,\ldots,\vec{S}_i\})$:
     $\vec{q}_i=\vec{S}_i-\sum_{j=1}^{i-1}\left(\vec{S}_i^T\vec{e}_j\right)\vec{e}_j$
  2. Normalize to compute $\vec{e}_i$: $\vec{e}_i=\frac{\vec{q}_i}{\|\vec{q}_i\|}$.

25.4 Implementing Gram Schmidt for OMP

Now, we would like to use Gram Schmidt to speed up OMP. Recall that in each iteration of OMP, we add one more device to our list of "on" devices, appending that song to a matrix: $A_j=[A_{j-1}\mid\vec{S}_j]$. Then, we use least squares to get the projection: $\vec{r}=A_j(A_j^TA_j)^{-1}A_j^T\vec{y}$. This step is the one that slows us down because we need to perform inversion.

The intuition is that instead of just appending the song $\vec{S}_j$ to the matrix A_{j-1}, we can use Gram-Schmidt as we go to make sure that A_j is orthonormal. Let Q_j be the orthonormal version of A_j. First, we initialize $Q_0=[\ ]$. Then, every time we find a new song, we perform Gram-Schmidt before adding it to Q, so Q remains orthonormal.

Recall that we want to solve for $\vec{x}$ in the following equation, where $\vec{x}$ has k nonzero entries:

$$A\vec{x} = \begin{bmatrix} | & | & & | \\ \vec{S}_1 & \vec{S}_2 & \dots & \vec{S}_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \sum_{j=1}^{n} x_j \vec{S}_j = \vec{y}.$$

In each step, we add one more song to our list of songs. Let $\vec{r}$ represent the residual, or the part of $\vec{y}$ not yet explained by our current list of songs. The detailed procedure is as follows:

Procedure:

- At time $j = 0$, we have no songs, so our residual is all of $\vec{y}$. So we initialize the following values: $\vec{r} = \vec{y}$, $Q = [\ ]$. We will also use a set T to hold the indices of the songs we have found so far, initialized to $\{\}$.
- Repeat for $j = 1, \dots, k$:
  1. Correlate $\vec{r}$ with all of the songs. Find the song $\vec{S}_{i_j}$ with the highest correlation. Add the song index i_j to the set T.
  2. Perform Gram Schmidt on Q and $\vec{S}_{i_j}$:
     (a) Find $\vec{q}_j$, or the part of $\vec{S}_{i_j}$ that is orthogonal to Q:
     $$\vec{q}_j = \vec{S}_{i_j} - \sum_{i=1}^{j-1} \left(\vec{S}_{i_j}^T \vec{e}_i \right) \vec{e}_i.$$
     (b) Find $\vec{e}_j$: $\vec{e}_j = \frac{\vec{q}_j}{\|\vec{q}_j\|}$
     (c) Column concatenate matrix Q with $\vec{e}_j$: $Q \leftarrow [Q \mid \vec{e}_j]$.
  3. To find the new residual, project $\vec{y}$ onto Q:
     $$\vec{r} = \vec{y} - \sum_{i=1}^{j} \left(\vec{y}^T \vec{e}_i \right) \vec{e}_i.$$
     We can speed this step up by noticing that in the previous time step, we had
     $$\vec{r} = \vec{y} - \sum_{i=1}^{j-1} \left(\vec{y}^T \vec{e}_i \right) \vec{e}_i.$$
     So we can compute the new residual by just subtracting the projection of $\vec{y}$ onto $\vec{e}_j$:
     $$\vec{r} \leftarrow \vec{r} - \vec{e}_j^T \vec{y}.$$

When the procedure terminates, we have a solution for $\vec{x}$ in the orthonormal Q basis. Let's denote this solution $\vec{x}^{(Q)}$, where $Q\vec{x}^{(Q)} = \vec{b}$. We still need to find $\vec{x}$ in the original basis of songs, or $A\vec{x} = \vec{b}$.

A slower solution is to use our set T: we have kept track of a set $T = \{i_1, i_2, \dots, i_k\}$. Then, we can form the matrix A_T whose columns are the songs we picked:

$$A_T = \begin{bmatrix} | & | & & | \\ \vec{S}_{i_1} & \vec{S}_{i_2} & \dots & \vec{S}_{i_k} \\ | & | & & | \end{bmatrix}$$

Let $\vec{x}_T$ be the entries of $\vec{x}$ corresponding to the indices in T. Then, we have

$$A_T \vec{x}_T = \begin{bmatrix} | & | & & | \\ \vec{S}_{i_1} & \vec{S}_{i_2} & \dots & \vec{S}_{i_k} \\ | & | & & | \end{bmatrix} \begin{bmatrix} x_{i_1} \\ x_{i_2} \\ \vdots \\ x_{i_k} \end{bmatrix} = \vec{y}.$$

The matrix A_T is dimension $m \times k$ where $k < m$, so we can solve for the non-zero entries of $\vec{x}$ using least squares.

Alternatively, we can use a change of basis: note that Gram-Schmidt gives us a way to map the original basis of songs $\{\vec{S}_{i_1}, \vec{S}_{i_2}, ..., \vec{S}_{i_k}\}$ to the orthonormal basis $\{\vec{e}_1, \vec{e}_2, ..., \vec{e}_k\}$. Our solution $\vec{x}^{(Q)}$ defines a linear combination of the orthonormal basis that reaches $\vec{b}$, or

$$Q\vec{x}^{(Q)} = \begin{bmatrix} | & | & & | \\ \vec{e}_1 & \vec{e}_2 & \dots & \vec{e}_k \\ | & | & & | \end{bmatrix} \begin{bmatrix} x_1^{(Q)} \\ x_2^{(Q)} \\ \vdots \\ x_k^{(Q)} \end{bmatrix} = \sum_{i=1}^{k} x_i^{(Q)} \vec{e}_i.$$

Suppose we have a change of basis matrix R such that

$$A_T = QR.$$

Then,

$$A_T \vec{x}_T = QR\vec{x}_T = Q\vec{x}^{(Q)},$$

Note that Q has full column rank, so it has no nullspace. Therefore, we can find $\vec{x}_T$ uniquely by

$$\vec{x}_T = R^{-1}\vec{x}^{(Q)}.$$

You might wonder how using the change of basis R is more efficient than using the index set T and least squares. The reason why it is more efficient is because R is upper triangular, meaning that all of its terms below the diagonal are 0, which makes inverting it very fast. If you want to learn more about this topic, take EE 127 or read about the QR Decomposition.

25.5 Practice Problems

These practice problems are also available in an interactive form on the course website (`http://ee16a.com/hw-practice/`).

1. True or False: We can apply Gram-Schmidt orthogonalization to any set of vectors or to the columns of any matrix to make the vectors or columns orthogonal and nonzero to each other.
2. Given the matrix $\begin{bmatrix} 3 & -1 \\ 2 & 5 \end{bmatrix}$, perform Gram-Schmidt orthonormalization on its columns. What is the resulting matrix?
3. How could we check if our result from Gram-Schmidt orthogonalization is correct?

(a) By taking the dot product of the original vector with the new Gram-Schmidt vector and seeing if it's 0.

(b) By checking if the dot product of each pair of the new Gram-Schmidt vectors is 0.

(c) By making sure the elements of the vector sum to 1.

4. True or False: The following code $\begin{bmatrix} 1 & -1 & 0 & 1 \end{bmatrix}$ is orthogonal to all (non-zero) circular shifts of itself.

5. True or False: There exists a pair of codes $\vec{c}_1, \vec{c}_2$ of length N and norm 1 such that $\vec{c}_1, \vec{c}_2$ are orthogonal to all non-zero circular shifts of themselves and to all circular shifts of each other.

6. True or False: Let $\mathbf{A}$ be a square matrix with orthonormal columns. Then $\mathbf{A}^{-1} = \mathbf{A}^T$.