EECS 16A Designing Information Devices and Systems I
Spring 2018 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 that in each iteration of the algorithm, first we found one 'on' device from the strongest signal component. Then we computed the projection of the measurement vector $\vec{r}$ onto the subspace spanned by the songs from 'on' devices found so far. Finally, we calculated the residual and used this to look for the next strongest song in the next iteration. If we let A_j be a matrix whose columns are the 'songs' found so far in iteration j and let $\vec{r}$ be the received signal, we compute $A_j \left(A_j^T A_j\right)^{-1} A_j^T \vec{r}$ to get the projection of $\vec{r}$ onto span(A_j). Matrix inversion and matrix multiplication can be computationally expensive - inversion is O(n^3) and multiplication is O(n^2). Is there a way to avoid doing such computations? Yes! It turns out that if the columns of A_j are mutually orthogonal to each other, the projection of $\vec{r}$ onto span(A_j) is the sum of the projection of $\vec{r}$ onto each of the columns of A_j. Recall that the projection of a vector $\vec{r}$ on to any other nonzero vector $\vec{b}$ of the same size is

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

This is fast to compute since computing the dot product of two vectors and the norm of a vector is linear in time in the number of components in the vectors.

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., $A_2 = \begin{bmatrix} | & | \\ \vec{S_1} & \vec{S_2} \\ | & | \end{bmatrix}$. Since $\vec{S_1}$ and $\vec{S_2}$ are orthogonal to each other, we have $\vec{S_1}^T \vec{S_2} = 0$. The least squares solution of $A_2 \vec{x} = \vec{r}$ is given by:

$$\hat{\vec{x}} = \left(A_2^T A_2\right)^{-1} A_2^T \vec{r}. \tag{2}$$

and we now multiply by A_2 to obtain the projection of the least sqaures solution onto the subspace spanned by A_2:

$$\vec{r}_{A_2} = A_2 \left(A_2^T A_2\right)^{-1} A_2^T \vec{r}. \tag{3}$$

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

$$A_2^T A_2 = \begin{bmatrix} — & \vec{S_1}^T & — \\ — & \vec{S_2}^T & — \end{bmatrix} \begin{bmatrix} | & | \\ \vec{S_1} & \vec{S_2} \\ | & | \end{bmatrix} \tag{4}$$

$$= \begin{bmatrix} \vec{S_1}^T \vec{S_1} & \vec{S_1}^T \vec{S_2} \\ \vec{S_2}^T \vec{S_1} & \vec{S_2}^T \vec{S_2} \end{bmatrix} \tag{5}$$

$$= \begin{bmatrix} \left\|\vec{S_1}\right\|^2 & 0 \\ 0 & \left\|\vec{S_2}\right\|^2 \end{bmatrix}. \tag{6}$$

$$\tag{7}$$

Thus, we have

$$\left(A_2^T A_2\right)^{-1} = \begin{bmatrix} \frac{1}{\|\vec{S_1}\|^2} & 0 \\ 0 & \frac{1}{\|\vec{S_2}\|^2} \end{bmatrix}. \tag{8}$$

Then the projection of $\vec{r}$ onto span(A_2) is

$$\vec{r}_{A_2} = A_2 \left(A_2^T A_2\right)^{-1} A_2^T \vec{r} \tag{9}$$

$$= \begin{bmatrix} | & | \\ \vec{S_1} & \vec{S_2} \\ | & | \end{bmatrix} \begin{bmatrix} \frac{1}{\|\vec{S_1}\|^2} & 0 \\ 0 & \frac{1}{\|\vec{S_2}\|^2} \end{bmatrix} \begin{bmatrix} — & \vec{S_1}^T & — \\ — & \vec{S_2}^T & — \end{bmatrix} \vec{r} \tag{10}$$

$$= \begin{bmatrix} | & | \\ \vec{S_1} & \vec{S_2} \\ | & | \end{bmatrix} \begin{bmatrix} \frac{1}{\|\vec{S_1}\|^2} & 0 \\ 0 & \frac{1}{\|\vec{S_2}\|^2} \end{bmatrix} \begin{bmatrix} \vec{S_1}^T \vec{r} \\ \vec{S_2}^T \vec{r} \end{bmatrix} \tag{11}$$

$$= \begin{bmatrix} | & | \\ \vec{S_1} & \vec{S_2} \\ | & | \end{bmatrix} \begin{bmatrix} \frac{\vec{S_1}^T \vec{r}}{\|\vec{S_1}\|^2} \\ \frac{\vec{S_2}^T \vec{r}}{\|\vec{S_2}\|^2} \end{bmatrix} \tag{12}$$

$$= \left(\frac{\vec{S_1}^T \vec{r}}{\left\|\vec{S_1}\right\|^2} \right) \vec{S_1} + \left(\frac{\vec{S_2}^T \vec{r}}{\left\|\vec{S_2}\right\|^2} \right) \vec{S_2}. \tag{13}$$

Observe that the first term in the sum above is the projection of $\vec{r}$ onto $\vec{S_1}$ and the second term is the projection of $\vec{r}$ onto $\vec{S_2}$. Generalizing, the projection of $\vec{r}$ onto span(A_n) where A_n has mutually orthogonal columns is

$$\vec{r}_{A_n} = \left(\frac{\vec{S_1}^T \vec{r}}{\left\|\vec{S_1}\right\|^2} \right) \vec{S_1} + \left(\frac{\vec{S_2}^T \vec{r}}{\left\|\vec{S_2}\right\|^2} \right) \vec{S_2} + \cdots + \left(\frac{\vec{S_n}^T \vec{r}}{\left\|\vec{S_n}\right\|^2} \right) \vec{S_n}. \tag{14}$$

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

$$\vec{r}_{A_n} = \left(\vec{S_1}^T \vec{r}\right) \vec{S_1} + \left(\vec{S_2}^T \vec{r}\right) \vec{S_2} + \cdots + \left(\vec{S_n}^T \vec{r}\right) \vec{S_n}, \tag{15}$$

which further reduces our computation.

We see that we can speed up OMP considerably if the songs $\vec{S}_1, \ldots, \vec{S}_n$ are mutually orthogonal to each other and are of unit length. Note: In this case, we cannot orthnormalize ALL of the songs $\vec{S}_1, \ldots \vec{S}_n$ because there are more songs than the length of the song (and therefore they cannot all be linearly independent). However, since we are dealing with a sparse system, in our example at most 10 devices transmit at once, we only have to orthonormalize some of the vectors. Now the question is how do we convert any set of linearly independent vectors to a set of mutually orthogonal unit vectors that span the same vector space? We will answer this in the next section.

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

Now what should α be if we would like $\vec{v}_1$ and $\vec{v}_2 - \alpha\vec{v}_1$ to be orthogonal to each other? We want $\alpha\vec{v}_1$ to be the projection of $\vec{v}_2$ onto $\vec{v}_1$. Let's solve this algebraically using the definition of orthogonality:

$$\vec{v}_1^T\left(\vec{v}_2 - \alpha\vec{v}_1\right) = 0 \tag{16}$$

$$\vec{v}_1^T\vec{v}_2 - \alpha\left\|\vec{v}_1\right\|^2 = 0 \tag{17}$$

$$\alpha = \frac{\vec{v}_1^T\vec{v}_2}{\left\|\vec{v}_1\right\|^2} \tag{18}$$

Definition 25.1 (Orthonormal): A set of vectors $\{\vec{S}_1, \ldots, \vec{S}_n\}$ 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 $\{q_1, \ldots, q_n\}$ that span the same vector space as the original set. Concretely, $\{q_1, \ldots, q_n\}$ needs to satisfy the following:

- span($\{S_1, \ldots, S_n\}$) = span($\{q_1, \ldots, q_n\}$)
- $\{q_1, \ldots, q_n\}$ 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 $\vec{q}_1$ such that span($\{\vec{q}_1\}$) = span($\{\vec{S}_1\}$).
  Since span($\{\vec{S}_1\}$) is a one dimensional vector space, the unit vector that span the same vector space would just be the normalized vector point at the same direction as $\vec{S}_1$. We have

$$\vec{q}_1 = \frac{\vec{S}_1}{\left\|\vec{S}_1\right\|}. \tag{19}$$

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

$$\vec{e}_2 = \vec{S}_2 - \left(\vec{S}_2^T\vec{q}_1\right)\vec{q}_1. \tag{20}$$

  Normalizing, we have $\vec{q}_2 = \frac{\vec{e}_2}{\|\vec{e}_2\|}$.

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

$$\left(\vec{S}_3^T\vec{q}_2\right)\vec{q}_2 + \left(\vec{S}_3^T\vec{q}_1\right)\vec{q}_1. \tag{21}$$

  We know that

$$\vec{e}_3 = \vec{S}_3 - \left(\vec{S}_3^T\vec{q}_2\right)\vec{q}_2 - \left(\vec{S}_3^T\vec{q}_1\right)\vec{q}_1 \tag{22}$$

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

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 $\{\vec{q}_1,\ldots,\vec{q}_n\}$, where span($\{\vec{S}_1,\ldots,\vec{S}_n\}$) = span($\{\vec{q}_1,\ldots,\vec{q}_n\}$).

Gram Schmidt Procedure

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

25.3 Implementing Gram Schmidt for OMP

Now we would like to use Gram Schmidt to speed up OMP. Recall the step in OMP that augments matrix A with the newest version of the song: $A = [A \mid \overrightarrow{S_i^{(Ni)}}]$ and then uses least squares to obtain the message value: $\vec{x} = (A^T A)^{-1} A^T \vec{r}$. We now know that with each iteration, that computation gets harder and harder.

Therefore, let us now build a matrix Q containing orthonormal vectors to use as a subspace rather than A. We can initialize $Q = [\ \ ]$ and once we identify a song, we can then perform Gram Schmidt on Q and the song.

In this case the received signal is $\vec{x}$ and the residual is $\vec{r}$.

Procedure:

- Initialize the following values: $\vec{r} = \vec{x}$, $j = 1$, $k = 10$, $F = \{\emptyset\}$, $Q = [\ \ ]$, $\vec{b_0} = 0$
- while $((j \leq k)$ & $(\| \vec{r} \| \geq th))$:
  1. Cross correlate $\vec{r}$ with the shifted versions of all songs. Find the song index, i, and the shifted version of the song, $\overrightarrow{S_i^{(Ni)}}$, with which the received signal has the highest correlation value.
  2. Set $\vec{v_j} = \overrightarrow{S_i^{(Ni)}}$
  3. Add i to the set of song indices, F.
  4. Perform Gram Schmidt on Q and $\vec{v_j}$
     (a) Find $\vec{e_j}$ which is $\vec{v_j}-$ projection of $\vec{v_j}$ onto Q : $\vec{e_j} = \vec{v_j} - \left(\vec{v_j}^T \vec{q_1}\right) \vec{q_1} - \cdots - \left(\vec{v_j}^T \vec{q}_{j-1}\right) \vec{q_{j-1}}$
     (b) Find $\vec{q_j}$: $\vec{q_j} = \frac{\vec{e_j}}{\|\vec{e_j}\|}$
     (c) Column concatenate matrix Q with $\vec{q_j}$: $Q = [Q \mid \vec{q_j}]$
  5. Now that vectors are orthonormal, can simply project received signal onto newest column and add: $\vec{b_j} = \vec{b_{j-1}} + \left(\vec{x}^T \vec{q_j}\right) \vec{q_j}$
  6. Update the residual value $\vec{r}$ by subtracting: $\vec{r} = \vec{x} - \vec{b_j}$
  7. Update the counter: $j = j + 1$