Greedy Algorithms

Objective: To find a polynomial-time algorithm to solve (P_0)

$$\boxed{\begin{aligned} &\min \|x\|_0 \\ &\text{s.t. } Ax = b \end{aligned} \qquad (P_0)}$$

Idea: One solution at a time!

you assume that $Ax=b$ can be satisfied.

Suppose that the solution contains only one non-zero. i.e. $\|x\|_0 = 1$, let's find this solution.

Denote a_j = j^{th} column of A.

Define

$$\mathcal{E}(j) = \min_{z_j} \underbrace{\| a_j z_j - b \|^2}_{\text{residue}}$$

z_j: scalar

Intuition: Find the best column that matches the observation.

$$\frac{\partial}{\partial z_j} \mathcal{E}(j) = \frac{\partial}{\partial z_j} \| a_j z_j - b \|^2$$

$$= 2 a_j^T (a_j z_j - b) = 0$$

$$\Rightarrow \quad z_j = \frac{a_j^T b}{\|a_j\|_2^2}$$

Therefore, $$\mathcal{E}(j) = \left\| a_j \left(\frac{a_j^T b}{\|a_j\|^2} \right) - b \right\|^2$$

$$= \frac{(a_j^T b)^2}{\|a_j\|^2} - 2\frac{(a_j^T b)^2}{\|a_j\|^2} + \|b\|^2$$

$$= \|b\|^2 - \frac{(a_j^T b)^2}{\|a_j\|^2}$$

Since $Ax=b$ has to be satisfied, and if $\|x\|_0=1$, then $\mathcal{E}(j)$ is able to reach $\mathcal{E}(j)=0$ for one of the $j=1,\dots,n$ columns. That is, j^* exists and

$$j^* = \arg\min_j \mathcal{E}(j).$$

The support of the solution is

$$S = \{j^*\}.$$

Correspondingly, the solution is

$$x^* = \arg\min_{x_S} \|A_S x_S - b\|^2 = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{a_{j^*}^T b}{\|a_{j^*}\|^2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

<u>Remark</u>: When $\mathcal{E}(j)=0$, we have

$$\mathcal{E}(j) = \|b\|^2 - \frac{(a_j^T b)^2}{\|a_j\|^2} = 0$$

$$\Longleftrightarrow (a_j^T b)^2 = \|a_j\|^2 \|b\|^2.$$

Since by Cauchy inequality:

$$(a_j^T b)^2 \le \|a_j\|^2 \|b\|^2,$$

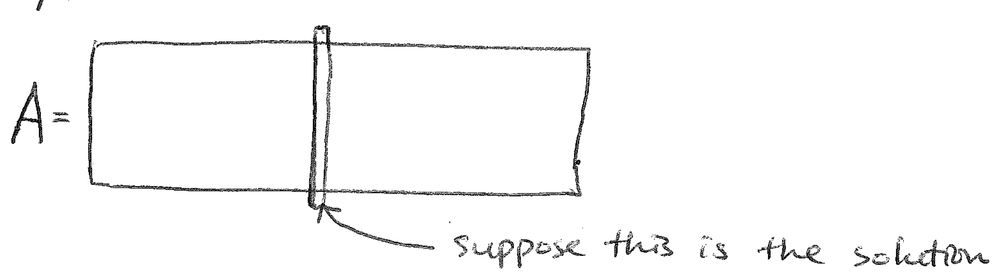
this implies that

$\mathcal{E}(j)=0$ when a_j is parallel to b.

Orthogonal Matching Pursuit

When $\|x\|_0 > 1$, the general procedure is to greedily pick up one non-zero at a time.

Initially, we have n columns.



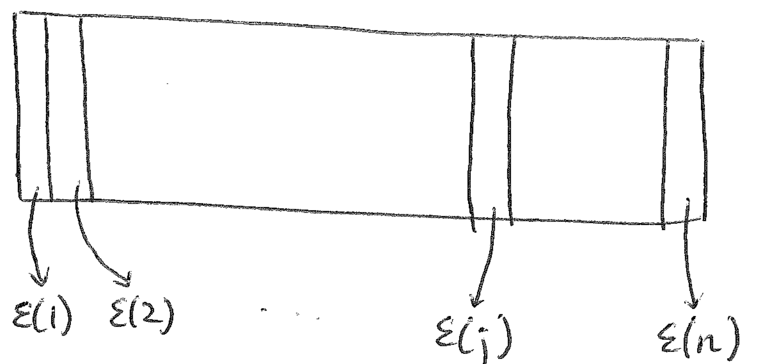
We define initial residue

$$r^0 \stackrel{\text{def}}{=} b.$$

Then, we compute the error associated with choose the j^{th} column

$$\varepsilon(j) \stackrel{\text{def}}{=} \min_{z_j} \| a_j z_j - r^0 \|^2$$

$$= \| r^0 \|^2 - \frac{(a_j^T r^0)^2}{\| a_j \|^2}$$

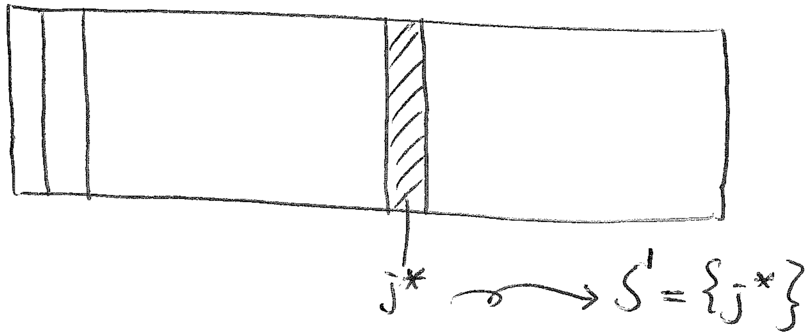


The best current j^* is the one that minimizes $\varepsilon(j)$

$$j^* = \arg\min_j \varepsilon(j)$$

Define the corresponding support

$$S^1 = \{j^*\}.$$



Then we solve a least squares problem

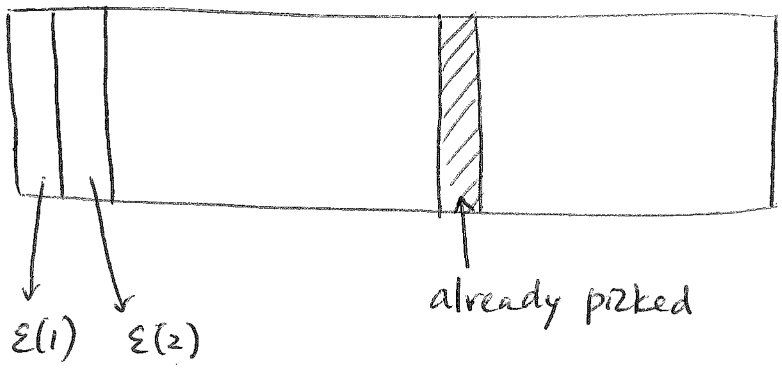
$$x^1 = \arg\min_{x_{S^1}} \| A_{S_1} x_{S_1} - b \|^2$$

$$= (A_{S_1}^T A_{S_1})^{-1} A_{S_1}^T b.$$

You always solve wrt b.

Now, when iteration $= 1$,

Define $r^1 = b - A_{S^1} x_{S^1}$ ← knock out the first solution



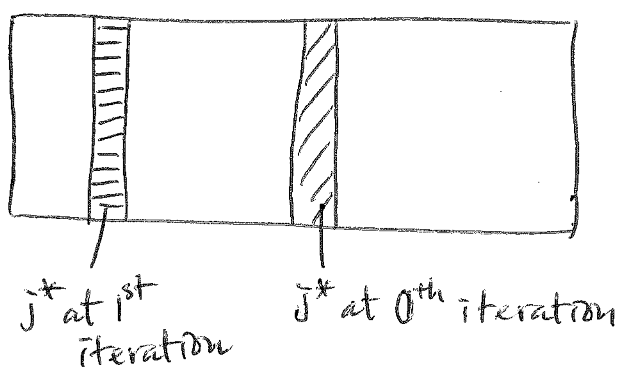
Define $\mathcal{E}(j) = \min_{z_j} \| a_j z_j - r^1 \|^2 = \|r^1\|^2 - \frac{(a_j^T r^1)^2}{\|a_j\|^2}$

The new

$$j^* = \underset{j \notin S^1}{\operatorname{argmin}} \ \mathcal{E}(j) \text{ , and then set}$$

$$S^2 = S^1 \cup \{ j^* \} .$$

~~The~~



So $$x^2 = \operatorname{argmin} \| A_{S^2} x_{S^2} - b \|^2$$

$$= (A_{S^2}^T A_{S^2})^{-1} A_{S^2}^T b .$$

In general, the above steps repeat ~~until~~ until you have pick k columns.

(1) ↑ something either you know ahead

(2) or when $Ax = b$ is solved.

Why called "orthogonal" matching pursuit?

Note that

$$\begin{cases} x^1 = (A_{S^1}^T A_{S^1})^{-1} A_{S^1}^T b \\ \text{and} \quad r^1 = b - A_{S^1} x_{S^1} . \end{cases}$$

Multiply $A_{S^1}^T$ to the second equation:

$$\begin{aligned} A_{S^1}^T r^1 &= A_{S^1}^T b - A_{S^1}^T A_{S^1} x_{S^1} \\ &= A_{S^1}^T b - (A_{S^1}^T A_{S^1})(A_{S^1}^T A_{S^1})^{-1} A_{S^1}^T b \\ &= 0 . \end{aligned}$$

So the columns of A_{S^1} are orthogonal to the residue r^1.

Matching Pursuit

"Problem" of orthognal matching pursuit:

no really a problem but a potential place to improve speed, with a cost.

When computing the least squares:

$$x^1 = \operatorname{argmin} \| A_{S^1} x_{S^1} - b \|^2 ,$$

we need to repeat this every time we update the support S. So the

question now is: Can we keep the values of x that are already computed?

Matching Pursuit Idea:

When forming $\mathcal{E}(j)$:

$$\mathcal{E}(j) = \min_{z_j} \| a_j z_j - r^{k-1} \|^2$$

we have $z_j^* = \dfrac{a_j^T r^{k-1}}{\| a_j \|^2}$

Therefore, $z_j^* = \dfrac{a_j^T r^{k-1}}{\|a_j\|^2}$ is the best coefficient for the current residue.

~~Difference~~

When updating the LS solution, we do

$$x^k = x^{k-1}$$
$$x^k(j^*) = x^{k-1}(j^*) + z_j^*.$$

Difference between OMP and MP:

OMP	MP
$x = \operatorname{argmin} \|A_S x_S - b\|^2$	$x^k(j^*) = x^{k-1}(j^*) + z_j^*$
update all coefficients in S simultaneously	update only the current coefficient.

The new residue for MP is

$$r^k = b - Ax^k$$
$$= r^{k-1} - z_j^* a_{j^*}$$

Remark: the residue for OMP is $r^k = b - Ax^k$.

Weak Matching Pursuit

Another "problem" of OMP:

$\varepsilon(j)$ needs to be computed until $Ax=b$ is satisfied. Question: can we stop earlier?

WMP:

$$\varepsilon(j) = \min_{z_j} \|a_j z_j - r^{k-1}\|^2$$
$$= \frac{a_j^T r^{k-1}}{\|a_j\|^2}.$$

Instead of keep adding j^* into S, WMP stops when

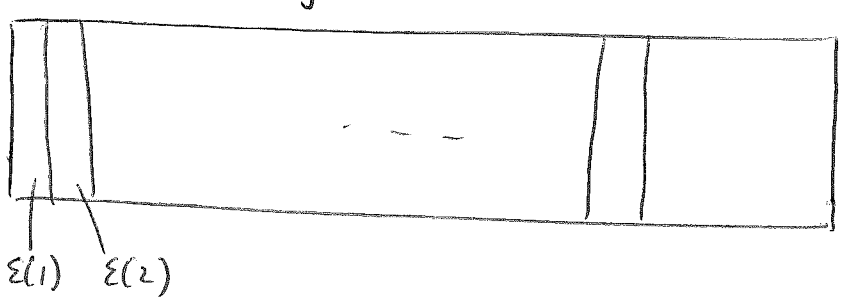
$$\cancel{\| a_j^T r^{k-1} \|}$$

$$\frac{| a_j^T r^{k-1} |}{\| a_j \|_2} \geq t \, \| r^{k-1} \|_2 ,$$ for some tolerance level t.

Thresholding

Directly pick the best k columns:

$$\varepsilon(j) = \min_{z_j} \| a_j z_j - b \|^2$$



Pick the smallest k $\varepsilon(j)$'s.

Then form the support set $S = \{ j_1^*, \dots, j_k^* \}$.

Solve $\min_{x_S} \| A_S x_S = b \|^2$.

Problem: Very weak algorithm

Advantage: Very cheap to compute.

Comparison of Greedy Algorithms

	Orthogonal Matching Pursuit	Matching Pursuit	Weak Matching Pursuit	Thresholding
Step 1	$\varepsilon(j) = \min_{z_j} \lVert a_j z_j - r^{k-1} \rVert^2$ $z_j^* = \frac{a_j^T r^{k-1}}{\lVert a_j \rVert^2}$			
Step 2	$j^* = \underset{j \notin S^{k-1}}{\operatorname{argmin}}\ \varepsilon(j)$ $S^k = S^{k-1} \cup \{j^*\}$		Stop when $\frac{\lvert a_j^T r^{k-1} \rvert}{\lVert a_j \rVert_2} \geq t \lVert r^{k-1} \rVert_2$	Pick the k columns with the smallest $\varepsilon(j)$.
Step 3	$x^k = \underset{x \in S}{\operatorname{argmin}} \lVert Ax - b \rVert^2$	$x^k = x^{k-1}$ $x^k(j^*) = x^{k-1}(j^*) + z_j^*$		$x^k = \underset{x \in S}{\operatorname{argmin}} \lVert Ax - b \rVert^2$
Step 4	$r^k = b - Ax^k$	$r^k = b - Ax^k$ $= r^{k-1} - z_j^* a_{j^*}$		not applicable

Least Squares OMP

OMP: $\min_{x \in S^k} \|Ax - b\|^2$,

and $\epsilon(j) = \min_{z_j} \| a_j z_j - r^k \|^2$.

LS-OMP: Solve both simultaneously:

$$\epsilon(j) = \min_{x_s,\, z} \left\| [A_s \,,\, a_j] \begin{bmatrix} x_s \\ z \end{bmatrix} - b \right\|^2$$

Solution: $\frac{\partial}{\partial x_s}(\cdot) = 0$ and $\frac{\partial}{\partial z}(\cdot) = 0$

$$\Rightarrow \cancel{[A_s^T \; a_j^T]} \begin{bmatrix} A_s^T \\ a_j^T \end{bmatrix} [A_s \; a_j] \begin{bmatrix} x_s \\ z \end{bmatrix} - \begin{bmatrix} A_s^T \\ a_j^T \end{bmatrix} b = 0$$

$$\Rightarrow \begin{bmatrix} x_s \\ z \end{bmatrix} = \begin{bmatrix} A_s^T A_s & A_s^T a_j \\ a_j^T A_s & \|a_j\|^2 \end{bmatrix}^{-1} \begin{bmatrix} A_s^T b \\ a_j^T b \end{bmatrix}.$$

The difference:

In OMP, the k^{th} estimate is found when all $k-1$ estimates are fixed. In LS-OMP, all k estimates are updated simultaneously.

In OMP, one has to solve a LS problem after the support set is determined. In LS-OMP, the support and the LS solution are found simultaneously.

Normalization

Question: Will normalization of columns of A affect the performance of OMP?

Let A be a matrix. Let $\tilde{A}$ be the column normalized matrix

$$\tilde{A} = AW$$

so that $\|\tilde{a}_j\|_2 = 1$.

Theorem (3.1)

The OMP produces the same support S^k when using either A or $\tilde{A}$.

Proof: Consider the sweep:

$$\begin{aligned}
\epsilon(j) &= \|r^{k-1}\|^2 - \left(\frac{a_j^T r^{k-1}}{\|a_j\|_2}\right)^2 \\
&= \|r^{k-1}\|^2 - (\tilde{a}_j^T r^{k-1})^2 \\
&= \|r^{k-1}\|^2 - \left(\frac{\tilde{a}_j^T r^{k-1}}{\|\tilde{a}_j\|_2}\right)^2 .
\end{aligned}$$

So using a_j or $\tilde{a}_j$ will lead to the same $\epsilon(j)$, and hence

$$j^* = \arg\min_j \epsilon(j)$$

is the same for both a_j and $\tilde{a}_j$.

For the LS step, since

$$x_S = (A_s^T A_s)^{-1} A_s^T b .$$

So

$$\begin{aligned} r^k &= b - A_s x_s \\ &= b - A_s (A_s^T A_s)^{-1} A_s^T b \\ &= \left(I - A_s (A_s^T A_s)^{-1} A_s^T \right) b \\ &= \left(I - A_s W_s W_s^{-1} (A_s^T A_s)^{-1} W_s^{-1} W_s A_s^T \right) b \\ &= \left(I - A_s W_s (W_s A_s^T A_s W_s)^{-1} W_s A_s^T \right) b \\ &= \left(I - \tilde{A}_s (\tilde{A}_s^T \tilde{A}_s)^{-1} \tilde{A}_s^T \right) b = \tilde{r}^k \end{aligned}$$

Therefore, the residue is unchanged.