EECS 16B Designing Information Devices and Systems II
Fall 2019 Discussion Worksheet

Discussion 7A

Outlier Removal via OMP

The problem of "outliers" (bad data points) is ubiquitous in the real world of data. This problem is about how we can leverage the techniques we already know to do something about them in a way that doesn't require a human to look at points and decide which ones are good or bad.

Suppose we have a system where we believe that vector-inputs $\vec{x}$ lead to scalar outputs in a linear way $\vec{p}^\top \vec{x}$. However, the parameters $\vec{p}$ are unknown and must be learned from data. Our data collection process is imperfect and instead of directly seeing $\vec{p}^\top \vec{x}$, we get observations $y = \vec{p}^\top \vec{x} + w$ where the w is some kind of disturbance or noise that nature introduces.

To help us learn the weights, we have n data points: input-output pairs $(\vec{x}_i, y_i)$ where the index $i = 1, \ldots, n$. However, because we believe that our observed outputs might be a bit noisy, we only have an approximate system of equations. In particular, if we group the data points into a matrix and vector as follows:

$$X = \begin{bmatrix} \vec{x}_1^\top \\ \vec{x}_2^\top \\ \vdots \\ \vec{x}_n^\top \end{bmatrix}$$

where clearly X is a matrix that has d columns and n rows. and

$$\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

is an n-vector. Then we can express the approximate system of equations that we want to solve as $X\vec{p} \approx \vec{y}$.

The classic least-squares problem views the goal as minimizing

$$\sum_{i=1}^{n} (y_i - \vec{x}_i^\top \vec{p})^2 \tag{1}$$

over the choice of $\vec{p}$ and thereby making the residual $\vec{y} - X\vec{p}$ have as small a Euclidean norm as possible. This is a reasonable thing to do when we believe that the residuals $y_i - \vec{x}_i^\top \vec{p}$ are small, meaning that we believe that the disturbance vector $\vec{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$ is small.

However, nature (or our observation process) isn't always this well behaved. When a small subset of observations don't line up well like the other data points, we call these *outliers*.

An exaggerated example is when we have a set of observations that satisfied perfectly $y_i = \sum_{j=1}^{d} x_{ij}$ with the $|y_i| < 1$ for all $i \neq c$, but there is one crazy x_c such that $y_c = \sum_{j=1}^{d} x_{cj} + 10^{100}$. Then, as we can see, if we were to do a standard least-squares solution that attempts to minimize (1), this single crazy observation would be enough to shift our estimated $\hat{\vec{p}}$ from the "true" $\vec{p}^* = [1, \ldots, 1]^\top$ by a reasonably large amount. Why? Because $\hat{\vec{p}} = (X^\top X)^{-1} X^\top \vec{y}$ is a linear function of $\vec{y}$ and hence the crazy huge deviation in the c-th component of $\vec{y}$ is going to cause a huge multiple of the c-th column of $(X^\top X)^{-1} X^\top$ to be added to the estimate for $\vec{p}$. The c-th column of $(X^\top X)^{-1} X^\top$ is just $(X^\top X)^{-1} \vec{x}_c$ by our definition of the matrix X above. And so, from the perspective of being an outlier that corrupts our estimate it really doesn't much matter whether the observation fault was in the scalar y_c or in the vector $\vec{x}_c$ — whichever side of the approximate equation representing the c-th data point is messed up, our estimate is going to be pretty badly messed up if we just blindly use least-squares.

Consequently, it is evident that the least-squares-estimated $\hat{p}$ is not what we really always want. Is there a way that allows us to reliably remove outliers when we know only a small proportion of the data points are outliers?

In this problem, we will demonstrate one of the simplest outlier removal methods that leverage the orthogonal matching pursuit (OMP) approach you learned in 16A.

Questions:

1. Assume for now we have a system of the form

$$\vec{y} \approx p\vec{x}.$$

What would be our estimate of p if we solved this through minimizing Eq. (1)? For this whole discussion, let's consider the most simple case where all of our x_i's are 1.

2. Following the above, **what would the estimation $\hat{p}$ be if one of the data points y_3 is corrupted by some noise that adds a constant R to the observed value?** i.e., the data point you observe is

$$\tilde{y}_3 = y_3 + R = px_3 + R.$$

You may assume that for the remaining data points x_j, you observe exactly px_j.

3. Following the previous parts, let's try to look at this problem in matrix form. **Write a least-squares expression in terms of vectors $\vec{x}$, $\vec{y}$, and scalar p. Then, find the solution $\hat{p}$ to the least-squares problem.**

4. As we saw in our example, we could improve our solution if we could remove outliers. One way to do this is by augmenting our matrices in the following way. Let $\vec{e}_i$ be the i-th standard basis vector. That is,

$$\vec{e}_i = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

where the solitary 1 is in the i-row of the vector $\vec{e}_i$. Consider the augmented matrix and vector

$$X_{new} = \begin{bmatrix} \vec{x} & \vec{e}_i \end{bmatrix}, \ \vec{y}_{new} = \vec{y}, \ \vec{p}_{new} = \begin{bmatrix} p \\ f_i \end{bmatrix}. \tag{2}$$

Apply the standard understanding of least squares to find the solution to

$$\min_{\vec{p}_{new}} \|\vec{y}_{new} - X_{new}\vec{p}_{new}\|_2^2. \tag{3}$$

and **write out the formula for the estimate** $\widehat{\vec{p}}_{new} = \begin{bmatrix} \widehat{p} \\ \widehat{f}_i \end{bmatrix}$. **How does this differ from the previous part?**

5. In the previous part, the 2×2 matrix $X_{new}^\top X_{new}$ played a role. Let's look at this matrix in "block" form:

$$X_{new}^\top X_{new} = \begin{bmatrix} \vec{x} & \vec{e}_i \end{bmatrix}^\top \begin{bmatrix} \vec{x} & \vec{e}_i \end{bmatrix} = \begin{bmatrix} \vec{x}^\top \\ \vec{e}_i^\top \end{bmatrix} \begin{bmatrix} \vec{x} & \vec{e}_i \end{bmatrix} = \begin{bmatrix} \vec{x}^\top\vec{x} & \vec{x}^\top\vec{e}_i \\ \vec{e}_i^\top\vec{x} & \vec{e}_i^\top\vec{e}_i \end{bmatrix}$$

What are $\vec{x}^\top\vec{e}_i, \vec{e}_i^\top\vec{x}, \vec{e}_i^\top\vec{e}_i$**?**

Simplify these as much as possible in terms of the individual data points $\vec{x}_i$, etc.

6. Based on what you know from the previous part, revisit the formula you got in the part before that and **prove that the value for** y_i **only impacts the estimate** $\widehat{f}_i$ **and does not effect the least-squares estimate for** $\widehat{\vec{p}}$ **at all.**

7. Argue that with the augmented variables defined in Eq. (2)**, the least-squares solution is equivalent to solving least-squares while ignoring the** i**-th set of data.**

8. Consider the following algorithm where we maintain a set of selected features A_{sel}. We use the fully augmented feature matrix

$$X = \begin{bmatrix} \vec{x} & I \end{bmatrix}$$

and try to solve for $p_{new} := \begin{bmatrix} p & f_1 & \dots & f_n \end{bmatrix}^\top$.

- Initialize residual $r = y$.
- Find $j := \arg\max X_i^\top r$, and add the j-th column X_j to A_{sel}. In other words, $A_{sel} \leftarrow \begin{bmatrix} A_{sel} & X_j \end{bmatrix}$.
- Calculate new residue $r \leftarrow r - A_{sel}(A_{sel}^\top A_{sel})^{-1}A_{sel}^\top r$.
- Repeat the above two procedures until we stop.

After we run this algorithm, we set $\hat{p} = \hat{p}_{new}[1]$ to be our estimate of p. **Discuss how the algorithm works. Does it make sense to run the algorithm until all features in** X **are selected?**

9. The above part implies that for our algorithm to work we would need to set a stopping condition. Intuitively this makes sense—after you remove the outliers, you end up just removing your naturally occurring extremes in your data. **What kinds of stopping conditions might be suitable for this problem?** This is an open problem and doesn't have a fixed solution.

Contributors:

- Jaijeet Roychowdhury.
- Aditya Arun.
- Kuan-Yun Lee.
- Anant Sahai.

