MATH 342W / 642 / RM 742 Spring 2025 HW #3

Sergio E. Garcia Tapia

Wednesday 19th March, 2025

Problem 1

These are questions about Silver's book, chapters 3-6. For all parts in this question, answer using notation from class (i.e. $t, f, g, h^*, \delta, \epsilon, e, t, z_1, \ldots, z_t, \mathbb{D}, \mathcal{H}, \mathcal{A}, \mathcal{X}, \mathcal{Y}, X, y, n, p, x_{\cdot 1}, \ldots, x_{\cdot p}$, $x_{1\cdot}, \ldots, x_{n\cdot}$, etc.

(a) [difficult] Chapter 4 is all about predicting weather. Broadly speaking, what is the problem with weather predictions? Make sure you use the framework and notation from class. This is not an easy question.

One problem is how computationally expensive it is. To predict the weather for a relatively large region, the atmosphere is broken down into a grid. A coarse grid may not be accurate, but the effect of using a finer grid exponentially increases the amount of computation that needs to be done. Put another way, for a large enough n, the algorithm $\mathcal{A}$ may take a long time to converge.

Another problem is the effect of nonlinearity and dynamic weather conditions, subsumed in the notion of chaos theory. The weather phenomenon is described by the unknown $t(z_1, z_2, \ldots, z_t)$, and the best we can do is decide on a model that assumes a functional form, captured by the hypothesis set $\mathcal{H}$. The price we pay to this point is ϵ, the noise, which includes both ignorance error and misspecification error. The effect of chaos theory is to accentuate this error over time, and thus predictions of the conditions far enough in the future (say, a week in advance) hardly beat the null model g_0.

(b) [easy] Why does the weatherman lie about the chance of rain? And where should you go if you want honest forecasts?

Weathermen lie because they are aware that consumers are likelier to notice certain kinds of mispredictions. Weather forecasts are probabilistic, communicating the likelihood that it might rain, for example, rather than deterministic statements about what actually will occur. Put another way, the response space $\mathcal{Y}$ is $[0, 1]$, which is numeric, to describe likelihood. However, consumers often use these forecasts to make deterministic decisions (yes or no to a picnic, for example), which for them is the binary response space $\mathcal{Y} = \{0, 1\}$. If the weatherman communicates a relatively high percentage of a small chance of something inconvenient occurring (such as rain), and it actually occurs,

consumers may be quick to criticize weathermen and lose trust in their forecasts. This is the reason for the "wet bias" that Silver describes. In the context of our framework, thus amounts to choosing an algorithm $\mathcal{A}$ that weighs certain features $x_{\cdot,k}$ heavier than others, particularly those associated with the chance of precipitation. Therefore, the forecasts made by weathermen often fail in the account of the "honesty" measure of predictions as described by Allan Murphy; they are intentionally not making the best forecast they can because of political reasons.

Instead of relying on weathermen, to obtain honest forecasts, it's best to go to the National Weather Service, as well as the Hurricane Center. These organizations strive for more quality (accuracy) and consistency (honesty).

(c) [difficult] Chapter 5 is all about predicting earthquakes. Broadly speaking, what is the problem with earthquake predictions? It is *not* the same as the problem of predicting weather. Read page 162 a few times. Make sure you use the framework and notation from class.

To create a model that we can use to predict a phenomenon, we need real data. In our framework, that corresponds to the true drivers $z_1, \ldots, z_t$ and the response y. Since we cannot hope to know the true drivers, we use proxies $x_1, \ldots, x_p$. However, we need to be able to measure the predictors $x_1, \ldots, x_p$ and their corresponding response y. In the case of earthquake predictions, there is no way to measure the predictors, such as the so-called "stress". Even if seismologists believe such features hold predictive power, they cannot verify it without measurement, and therefore they cannot justify making accurate predictions using them.

(d) [easy] Silver has quite a whimsical explanation of overfitting on page 163 but it is really educational! What is the nonsense predictor in the model he describes?

The "nonsense predictor" is the color, call it x_1, of the combination lock. Surely the color of the lock has little or nothing to do with how to pick it. That is, x_1 is not a good proxy to the true drivers $z_1, \ldots, z_t$, and it will be unable to reliable predict $\hat{y}_*$.

(e) [easy] John von Neumann was credited with saying that "with four parameters I can fit an elephant and with five I can make him wiggle his trunk". What did he mean by that and what is the message to you, the budding data scientist?

He meant that you can fit any curve you want to a data set, and manipulate the curve at will to make it appear valid. However, in doing so we are deluding ourselves and being dishonest, since we artificially increase p, SSR, and R^2 (and decrease SSE). This insight reminds us as data scientists to not get carried away and become vulnerable to overfitting, blindly matching a curve to fit the noise in the data, lest our resulting model will be garbage.

(f) [difficult] Chapter 6 is all about predicting unemployment, an index of macroeconomic performance of a country. Broadly speaking, what is the problem with unemployment predictions? It is *not* the same as the problem of predicting weather or earthquakes.

Make sure you use the framework and notation from class.

According to Hatzius, as discussed on page 185, the fundamental challenges are: first it is hard to determine the cause and effect from economic statistics alone, second the economy is always changing, and third the data available to economists is very noisy.

The predictors (indicators) for weather, the $x_1, \ldots, x_p$, can be measured, and since they are well-understood scientifically, they continue to be good indicators over time. Meanwhile, the indicators used to predict earthquakes cannot be measured. For the economy, an indicator that may be valid at one point in time may no longer be valid at a different point in time. In other words, we cannot claim that $x_1, \ldots, x_p$ is a good set of indicators, and then expect that they will continue to be, as for the weather; the set and number of the change over time. Using the language from today's class, the functions t, f, and h^* are not stationary.

The other problem is that the relationship between the economic variables changes over time.

(g) [E.C.] Many times in this chapter Silver says something on the order of "you need to have theories about how things function in order to make good predictions." Do you agree? Discuss.

Problem 2

These are questions related to the concept of orthogonal projection, QR decomposition and its relationship with least squares linear modeling.

(a) [easy] Let $\boldsymbol{H}$ be the orthogonal projection onto colsp $[\boldsymbol{X}]$ where $\boldsymbol{X}$ is a $n \times (p+1)$ matrix with all columns linearly independent from each other. What is rank $[\boldsymbol{H}]$?

rank $[\boldsymbol{H}] = p+1$. I will omit the proof because it is not requested, but here is the idea of the argument. We assume that $p+1 \leq n$, and since $\boldsymbol{X}$ has linearly independent columns, rank $[\boldsymbol{X}] = p+1$. We can then show that rank $[\boldsymbol{X}]$ = rank $\left[\boldsymbol{X}^\top \boldsymbol{X}\right]$. Further, since $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}$, we can show that nullspace$[\boldsymbol{X}^\top \boldsymbol{X}]$ = nullspace$[\boldsymbol{H}]$. Finally, using the fundamental theorem of linear maps (also known as the rank-nullity theorem), we can conclude that rank $[\boldsymbol{H}]$ = rank $\left[\boldsymbol{X}^\top \boldsymbol{X}\right]$ = rank $[\boldsymbol{X}] = p+1$.

(b) [easy] Simplify $\boldsymbol{H}\boldsymbol{X}$ by substituting for $\boldsymbol{H}$.

We have shown that $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top$. Therefore,

$$\begin{aligned}
\boldsymbol{H}\boldsymbol{X} &= (\boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top)\boldsymbol{X} \\
&= \boldsymbol{X}\underbrace{[(\boldsymbol{X}^\top \boldsymbol{X})^{-1}(\boldsymbol{X}^\top \boldsymbol{X})]}_{\boldsymbol{I}_{p+1}} \qquad \text{(by Associativity of Matrix Multiplication)} \\
&= \boldsymbol{X}\boldsymbol{I}_{p+1} \\
&= \boldsymbol{X}
\end{aligned}$$

(c) [harder] What does your answer from the previous question mean conceptually?

When the columns of $\boldsymbol{X}$ are projected onto the column space of $\boldsymbol{X}$, they remain unchanged. Put another way, the orthogonal projection matrix $\boldsymbol{H}$ is the identity when restricted to the column space of $\boldsymbol{X}$. Thus if $\boldsymbol{x}_{\cdot j}$ is the jth column of $\boldsymbol{X}$, we have

$$\boldsymbol{H}\boldsymbol{x}_{\cdot j} = \boldsymbol{x}_{\cdot j}, \quad \forall_{0 \le j \le p}$$

Yet another interpretation is that the columns of $\boldsymbol{X}$ are eigenvectors of $\boldsymbol{H}$, all corresponding to eigenvalue $\lambda = 1$.

(d) [difficult] Let $\boldsymbol{X}'$ be the matrix of $\boldsymbol{X}$ whose columns are in reverse order meaning that $\boldsymbol{X} = [\mathbf{1}_n \vdots \boldsymbol{x}_{\cdot 1} \vdots \ldots \vdots \boldsymbol{x}_{\cdot p}]$ and $\boldsymbol{X}' = [\boldsymbol{x}_{\cdot p} \vdots \ldots \vdots \boldsymbol{x}_{\cdot 1} \vdots \mathbf{1}_n]$. Show that the projection matrix that projects onto colsp $[X]$ is the same exact projection matrix that projects onto colsp $[X']$.

Proof. Let $\boldsymbol{H}$ be the orthogonal projection onto colsp $[\boldsymbol{X}]$, and let $\boldsymbol{X}'$ be the orthogonal projection onto colsp $[\boldsymbol{X}']$. By parts 2(b) and 2(c), we know $\boldsymbol{H}\boldsymbol{X} = \boldsymbol{X}$ and $\boldsymbol{H}'\boldsymbol{X}' = \boldsymbol{X}'$, and hence $\boldsymbol{H}\boldsymbol{x}_{\cdot j} = \boldsymbol{x}_{\cdot j}$ and $\boldsymbol{H}'\boldsymbol{x}'_{\cdot j} = \boldsymbol{x}'_{\cdot j}$, where $\boldsymbol{x}_{\cdot j}$ is the jth column of $\boldsymbol{X}$ and $\boldsymbol{x}'_{\cdot j}$ is the jth column of $\boldsymbol{X}'$. Since the set of columns of $\boldsymbol{X}$ and $\boldsymbol{X}'$ is the same (indeed, the list is merely written in reverse order), we have

$$\boldsymbol{H}\boldsymbol{x}_{\cdot j} = \boldsymbol{x}_{\cdot j} = \boldsymbol{H}'\boldsymbol{x}_{\cdot j}, \quad \forall_{0 \le j \le p}$$

and hence

$$\boldsymbol{H}\boldsymbol{X} = \boldsymbol{X} = \boldsymbol{H}'\boldsymbol{X}$$

Note colsp $[\boldsymbol{X}]$ = colsp $[\boldsymbol{X}']$ because $\boldsymbol{X}$ and $\boldsymbol{X}'$ have the same set of columns, albeit in different order. Let $U :=$ colsp $[\boldsymbol{X}]$ = colsp $[\boldsymbol{X}']$. Then $U^\top$ is the orthogonal complement of U, and it has dimension $n - (p+1)$ since

$$\boldsymbol{R}^n = U \oplus U^\top$$

where $\oplus$ denotes a direct sum of vector spaces. Let $\boldsymbol{x}_{\cdot 1,\perp} \ldots, \boldsymbol{x}_{\cdot n-(p+1),\perp}$ be a basis for $U^\perp$. Since $\boldsymbol{H}$ and $\boldsymbol{H}'$ are orthogonal projections onto U, it follows that

$$\boldsymbol{H}\boldsymbol{x}_{\cdot k,\perp} = \mathbf{0}_n = \boldsymbol{H}'\boldsymbol{x}_{\cdot k,\perp}, \quad \forall_{(p+1) \le k \le n}$$

Since the vectors in the list $\boldsymbol{x}_{\cdot 1,\perp} \ldots, \boldsymbol{x}_{\cdot n-(p+1),\perp}$ is orthogonal to the linearly independent list $\boldsymbol{x}_{\cdot 0}, \boldsymbol{x}_{\cdot 1}, \ldots, \boldsymbol{x}_{\cdot p}$, altogether they make up a basis of $\boldsymbol{R}^n$. At this, we have concluded that $\boldsymbol{H}$ and $\boldsymbol{H}'$ map a basis of $\boldsymbol{R}^n$ in precisely the same way, so this is sufficient to conclude that $\boldsymbol{H} = \boldsymbol{H}'$. However, we can be more expicit. Let's extend $\boldsymbol{X}$ to $n \times n$ matrix $\boldsymbol{X}_{\text{full}}$ by appending the basis vectors of $U^\perp$:

$$\boldsymbol{X}_{\text{full}} = \begin{bmatrix} \uparrow & \uparrow & \cdots & \uparrow & \uparrow & \cdots & \uparrow \\ \mathbf{1}_n & \boldsymbol{x}_{\cdot 1} & \cdots & \boldsymbol{x}_{\cdot p} & \boldsymbol{x}_{\cdot 1,\perp} & \cdots & \boldsymbol{x}_{\cdot n-(p+1),\perp} \\ \downarrow & \downarrow & \cdots & \downarrow & \downarrow & \cdots & \downarrow \end{bmatrix}$$

Thus,

$$\begin{aligned} \boldsymbol{H}\boldsymbol{X}_{\text{full}} &= [\boldsymbol{H}\boldsymbol{X} \mid \boldsymbol{H}\boldsymbol{0}_{n\times(n-(p+1))}] \\ &= [\boldsymbol{H}'\boldsymbol{X} \mid \boldsymbol{H}'\boldsymbol{0}_{n\times(n-(p+1))}] \\ &= \boldsymbol{H}'\boldsymbol{X}_{\text{full}} \end{aligned}$$

Since the columns of $\boldsymbol{X}_{\text{full}}$ form a basis of $\boldsymbol{R}^n$, the matrix $\boldsymbol{X}_{\text{full}}$ is invertible. Multiplying by its inverse, we arrive at

$$\begin{aligned} \boldsymbol{H}\boldsymbol{X}_{\text{full}} &= \boldsymbol{H}'\boldsymbol{X}_{\text{full}} \\ \boldsymbol{H}\boldsymbol{X}_{\text{full}}(\boldsymbol{X}_{\text{full}})^{-1} &= \boldsymbol{H}'\boldsymbol{X}_{\text{full}}(\boldsymbol{X}_{\text{full}})^{-1} \\ \boldsymbol{H}\boldsymbol{I}_n &= \boldsymbol{H}'\boldsymbol{I}_n \\ \boldsymbol{H} &= \boldsymbol{H}' \end{aligned}$$

□

(e) [easy] Generalize the previous problem by proving that orthogonal projection matrices that project onto any specific subspace are *unique*.

Proof. Suppose that V is a finite-dimensional vector space, and U is a subspace of V. Let $\mathbf{u}_1, \ldots, \mathbf{u}_n$ be a basis for U, and let $\mathbf{u}_{1,\perp}, \ldots, \mathbf{u}_{m,\perp}$ be a basis of $U^\perp$. If P and P' are both orthogonal projections onto U, then $P\mathbf{u}_j = \mathbf{u}_j = P'\mathbf{u}_j$ for all $1 \leq j \leq n$, and $P\mathbf{u}_{k,\perp} = \mathbf{0}_n = P'\mathbf{u}_{k,\perp}$ for all $1 \leq k \leq m$.

Let A be a matrix whose columns consists of the vectors that are a basis of U followed by the vectors that are a basis of $U^\perp$:

$$A = \begin{bmatrix} \uparrow & \cdots & \uparrow & \uparrow & \cdots & \uparrow \\ \mathbf{u}_1 & \cdots & \mathbf{u}_n & \mathbf{u}_{1,\perp} & \cdots & \mathbf{u}_{m,\perp} \\ \downarrow & \cdots & \downarrow & \downarrow & \cdots & \downarrow \end{bmatrix}$$

Then A is invertible because its columns form a basis for V, given that we can decompose V as a direct sum $V = U \oplus U$. Then

$$PA = P \begin{bmatrix} \uparrow & \cdots & \uparrow & \uparrow & \cdots & \uparrow \\ \mathbf{u}_1 & \cdots & \mathbf{u}_n & \mathbf{u}_{1,\perp} & \cdots & \mathbf{u}_{m,\perp} \\ \downarrow & \cdots & \downarrow & \downarrow & \cdots & \downarrow \end{bmatrix}$$

$$
\begin{aligned}
&= \begin{bmatrix} \uparrow & \cdots & \uparrow & \uparrow & \cdots & \uparrow \\ P\mathbf{u}_1 & \cdots & P\mathbf{u}_n & P\mathbf{u}_{1,\perp} & \cdots & P\mathbf{u}_{m,\perp} \\ \downarrow & \cdots & \downarrow & \downarrow & \cdots & \downarrow \end{bmatrix} \\
&= \begin{bmatrix} \uparrow & \cdots & \uparrow & \uparrow & \cdots & \uparrow \\ P'\mathbf{u}_1 & \cdots & P'\mathbf{u}_n & P'\mathbf{u}_{1,\perp} & \cdots & P'\mathbf{u}_{m,\perp} \\ \downarrow & \cdots & \downarrow & \downarrow & \cdots & \downarrow \end{bmatrix} \\
&= P'A
\end{aligned}
$$

Since A is invertible, we have

$$P = PI = P(AA^{-1}) = (PA)A^{-1} = (P'A)A^{-1} = P'(AA^{-1}) = P'I = P'$$

where I is the identity matrix on V. Hence, $P = P'$. □

(f) [easy] Prove that if a square matrix is both symmetric and idempotent then it must be an orthogonal projection matrix.

Proof. Suppose that P is an $n \times n$ matrix that is both symmetric ($P = P^\top$) and idempotent ($P^2 = P$). Let $\mathbf{u}, \mathbf{v}$ belong to $\mathbb{R}^n$. Then

$$
\begin{aligned}
(\mathbf{u} - P\mathbf{u})^\top (P\mathbf{v}) &= (\mathbf{u}^\top - (P\mathbf{u})^\top)(P\mathbf{v}) \\
&= (\mathbf{u}^\top - \mathbf{u}^\top P^\top)(P\mathbf{v}) \\
&= \mathbf{u}^\top P\mathbf{v} - \mathbf{u}^\top P^\top P\mathbf{v} \\
&= \mathbf{u}^\top P\mathbf{v} - \mathbf{u}^\top P \cdot P\mathbf{v} && \text{(Symmetry: } P = P^\top\text{)} \\
&= \mathbf{u}^\top P\mathbf{v} - \mathbf{u}^\top P\mathbf{v} && \text{(Idempotency: } P^2 = P\text{)} \\
&= 0
\end{aligned}
$$

Therefore, P is an orthogonal projection matrix. □

(g) [difficult] [MA] Prove the converse of the previous question: that if a square matrix is an orthogonal projection matrix, then it must be both symmetric and idempotent.

Proof. Suppose P an $n \times n$ orthogonal projection matrix. Let $\mathbf{v} \in \mathbb{R}^n$, and define

$$\mathbf{u} := P^\top \mathbf{v} - P\mathbf{v}$$

If $\mathbf{w} \in \mathbb{R}^n$, then

$$\mathbf{w}^\top \mathbf{u} = \mathbf{w}^\top P^\top \mathbf{v} - \mathbf{w}^\top P\mathbf{v}$$

Since P is an orthogonal projection matrix, it follows that

$$\begin{aligned} 0 &= (\mathbf{w} - P\mathbf{w})^\top (P\mathbf{v}) \\ &= \mathbf{w}^\top P\mathbf{v} - \mathbf{w}^\top P^\top P\mathbf{v} \end{aligned}$$

= Now we have

$$\begin{aligned} \mathbf{w}^\top \mathbf{u} &= \mathbf{w}^\top P^\top \mathbf{v} - \mathbf{w}^\top P^\top P\mathbf{v} \\ &= \mathbf{w}^\top P^\top (\mathbf{v} - P\mathbf{v}) \\ &= (P\mathbf{w})^\top (\mathbf{v} - P\mathbf{v}) \\ &= 0 \qquad\qquad \text{(since } P \text{ is an orthogonal projection matrix)} \end{aligned}$$

Since $\mathbf{w}$ is arbitrary, this means $\mathbf{u}$ is orthogonal to every vector in $\mathbb{R}^n$, and hence it must be the zero vector. Now

$$\mathbf{0}_n = P^\top \mathbf{v} - P\mathbf{v} \quad \implies \quad P^\top \mathbf{v} = P\mathbf{v}$$

Since $\mathbf{v}$ is also arbitrary, this implies $P^\top = P$, so P is idempotent. Using a similar argument, we can show that it is symmetric. Let $\mathbf{v} \in \mathbb{R}^n$, and define

$$\mathbf{a} := P^2\mathbf{v} - P\mathbf{v}$$

Given any vector $\mathbf{b} \in \mathbb{R}^n$, we get

$$\begin{aligned} \mathbf{b}^\top \mathbf{a} &= b^\top P^2 \mathbf{v} - b^\top P\mathbf{v} \\ &= b^\top P \cdot P\mathbf{v} - b^\top P\mathbf{v} \\ &= b^\top P^\top P\mathbf{v} - b^\top P^\top \mathbf{v} \qquad\qquad \text{(Symmetry: } P = P^\top\text{)} \\ &= b^\top P^\top (P\mathbf{v} - \mathbf{v}) \\ &= (P\mathbf{b})^\top (P\mathbf{v} - \mathbf{v}) \\ &= 0 \qquad\qquad \text{(since } P \text{ is an orthogonal projection matrix)} \end{aligned}$$

Since $\mathbf{b}$ is arbitrary, this means $\mathbf{a}$ is orthogonal to every vector in $\mathbf{R}^n$, and hence $\mathbf{a}$ is the zero vector. Thus $P^2\mathbf{v} = P\mathbf{v}$ for arbitrary $\mathbf{v}$, and hence, $P^2 = P$.

□

(h) [easy] Prove that I_n is an orthogonal projection matrix $\forall n$.

Proof. Given any $\mathbf{u}$ and $\mathbf{v}$ in $\mathbb{R}^n$,

$$(\mathbf{u} - I\mathbf{u})^\top (I\mathbf{v}) = (\mathbf{u} - \mathbf{u})^\top (I\mathbf{v}) = \mathbf{0}^\top (\mathbf{v}) = 0$$

Alternatively, note that $I^\top = I$ and $I^2 = I$, so by part (f), we see I is an orthogonal projection matrix. □

(i) [easy] What subspace does I_n project onto?

$\mathbb{R}^n$. Let U be the vector space that $\boldsymbol{I}_n$ is an orthogonal projection for. Let $\mathtt{w} \in U^\perp$, where $U^\perp$ is the orthogonal complement of U. Then $\boldsymbol{I}_n w = 0$. Therefore $U^\perp \subseteq \{0\}$. Since $U^\perp$ is a vector space, it cannot be empty, so $U^\perp = \{0\}$. Since $U^\perp$ is a subspace of $\boldsymbol{R}^n$, which is a finite-dimensional vector space, it follows that

$$U = (U^\perp)^\perp = \{\mathbf{0}_n\}^\perp = \boldsymbol{R}^n$$

The last equality follows because every vector in $\boldsymbol{R}^n$ is orthogonal to the zero vector $\mathbf{0}_n$.

(j) [easy] Consider least squares linear regression using a design matrix X with rank $p+1$. What are the degrees of freedom in the resulting model? What does this mean?

The number of degrees of freedom is $p + 1$. The degrees of freedom correspond to the p independent features together with the bias (intercept) term.

(k) [difficult] We saw that the perceptron is an *iterative algorithm*. This means that it goes through multiple iterations in order to converge to a closer and closer $\boldsymbol{w}$. Why not do the same with linear least squares regression? Consider the following. Regress $\boldsymbol{y}$ using $\boldsymbol{X}$ to get $\hat{\boldsymbol{y}}$. This generates residuals $\boldsymbol{e}$ (the leftover piece of $\boldsymbol{y}$ that wasn't explained by the regression's fit, $\hat{\boldsymbol{y}}$). Now try again! Regress $\boldsymbol{e}$ using $\boldsymbol{X}$ and then get new residuals $\boldsymbol{e}_{new}$. Would $\boldsymbol{e}_{new}$ be closer to $\mathbf{0}_n$ than the first $\boldsymbol{e}$? That is, wouldn't this yield a better model on iteration #2? Yes/no and explain.

The first time we apply regression, the least square estimates are given by

$$\boldsymbol{b} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

The prediction is $\hat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{b}$, and the residual is $\boldsymbol{e} := \hat{\boldsymbol{y}} - \boldsymbol{y}$. Next, we compute $\boldsymbol{b}_{new}$ by regressing on $\boldsymbol{e}$:

$$\begin{aligned} \boldsymbol{b}_{new} &= \boldsymbol{X}\boldsymbol{e} \\ &= (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top (\boldsymbol{y} - \hat{\boldsymbol{y}}) \\ &= (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}) \end{aligned}$$

$$
\begin{aligned}
&= (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y} - \underbrace{(\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{X}}_{\boldsymbol{I}_{p+1}} \boldsymbol{b} \\
&= \boldsymbol{b} - \boldsymbol{I}_{p+1} \boldsymbol{b} \\
&= \boldsymbol{0}_{p+1}
\end{aligned}
$$

Thus, $\hat{\boldsymbol{y}}_{new} = \boldsymbol{X}\boldsymbol{b}_{new} = \boldsymbol{X}\boldsymbol{0}_{p+1} = \boldsymbol{0}_n$, and $\boldsymbol{e}_{new} = \boldsymbol{y} - \hat{\boldsymbol{y}}_{new} = \boldsymbol{y}$, which is not closer to $\boldsymbol{0}_n$ at all.

(l) [harder] Prove that $\boldsymbol{Q}^\top = \boldsymbol{Q}^{-1}$ where $\boldsymbol{Q}$ is an orthonormal matrix such that $\text{colsp}\,[\boldsymbol{Q}] = \text{colsp}\,[\boldsymbol{X}]$ and $\boldsymbol{Q}$ and $\boldsymbol{X}$ are both matrices $\in \mathbb{R}^{n \times (p+1)}$ and $n = p + 1$ in this case to ensure the inverse is defined. Hint: this is purely a linear algebra exercise and it's a one-liner.

Proof. Let $\mathbf{q}_1, \ldots, \mathbf{q}_n$ be the columns of $\boldsymbol{Q}$, which form an orthonormal basis of $\mathbb{R}^n$. Then the ith row of $\boldsymbol{Q}^\top$ is precisely $\mathbf{q}_i^\top$, and

$$
(\boldsymbol{Q}^\top \boldsymbol{Q})_{ij} = \sum_{k=1}^{n} (\mathbf{q}_i^\top)_k (\mathbf{q}_j)_k = \mathbf{q}_i^\top \mathbf{q}_j = \begin{bmatrix} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{bmatrix}
$$

Hence $\boldsymbol{Q}^\top \boldsymbol{Q} = I_{p+1}$, so $\boldsymbol{Q}^\top = \boldsymbol{Q}^{-1}$. □

(m) [easy] Prove that the least squares projection $\boldsymbol{H} = \boldsymbol{X}\left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T = \boldsymbol{Q}\boldsymbol{Q}^\top$. Justify each step.

Let $\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{R}$ be a QR-decomposition of $\boldsymbol{X}$. Here, $\boldsymbol{Q}$ is an orthonormal matrix whose columns span exactly the same subspace as the columns of $\boldsymbol{X}$, and $\boldsymbol{R}$ is a square upper-triangle matrix. We further assume that $\boldsymbol{X}$ is full rank, so $\boldsymbol{R}$ is also full rank and hence invertible, as well as $\boldsymbol{R}^\top$. Now

$$
\begin{aligned}
H &= \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \\
&= (\boldsymbol{Q}\boldsymbol{R})((\boldsymbol{Q}\boldsymbol{R})^\top \boldsymbol{Q}\boldsymbol{R})^{-1} (\boldsymbol{Q}\boldsymbol{R})^\top \\
&= \boldsymbol{Q}\boldsymbol{R}(\boldsymbol{R}^\top \boldsymbol{Q}^\top \boldsymbol{Q}\boldsymbol{R})^{-1} \boldsymbol{R}^\top \boldsymbol{Q}^\top && \text{(since } (AB)^\top = B^\top A^\top\text{)} \\
&= \boldsymbol{Q}\boldsymbol{R}(\boldsymbol{R}^\top \boldsymbol{R})^{-1} \boldsymbol{R}^\top \boldsymbol{Q}^\top && \text{(since } \boldsymbol{Q}^\top = \boldsymbol{Q}^{-1}\text{)} \\
&= \boldsymbol{Q} \underbrace{\boldsymbol{R}\boldsymbol{R}^{-1}}_{\boldsymbol{I}_{p+1}} \underbrace{(\boldsymbol{R}^\top)^{-1} \boldsymbol{R}^\top}_{\boldsymbol{I}_{p+1}} \boldsymbol{Q}^\top && \text{(since } (AB)^{-1} = B^{-1}A^{-1}\text{)} \\
&= \boldsymbol{Q}\boldsymbol{Q}^\top
\end{aligned}
$$

(n) [difficult] [MA] This problem is independent of the others. Let H be an orthogonal projection matrix. Prove that $\text{rank}\,[\boldsymbol{H}] = \text{tr}\,[\boldsymbol{H}]$. Hint: you will need to use facts about eigenvalues and the eigendecomposition of projection matrices.

(o) [harder] Prove that an orthogonal projection onto the colsp $[\boldsymbol{Q}]$ is the same as the sum of the projections onto each column of $\boldsymbol{Q}$.

Proof. The orthogonal projection matrix onto the column space of $\boldsymbol{Q}$ is computed by

$$\boldsymbol{H} = \boldsymbol{Q}(\boldsymbol{Q}^\top \boldsymbol{Q})^{-1}\boldsymbol{Q}^\top = \boldsymbol{Q}(\boldsymbol{I}_{p+1})^{-1}\boldsymbol{Q}^\top = \boldsymbol{Q}\boldsymbol{Q}^\top$$

Suppose $\boldsymbol{Q}$ is $n \times (p+1)$. Let $\mathbf{q}_1, \ldots, \mathbf{q}_{p+1}$ be the columns of $\boldsymbol{Q}$, which form an orthonormal list. Also, let

$$\mathbf{q}_j := \begin{bmatrix} q_{1,j} \\ \vdots \\ q_{n,j} \end{bmatrix}$$

Then the projection matrix onto the jth column of $\boldsymbol{Q}$ is given by

$$\begin{aligned} \mathbf{q}_j \mathbf{q}_j^\top &= \mathbf{q}_j \begin{bmatrix} q_{1,j} & \cdots & q_{n,j} \end{bmatrix} \\ &= \begin{bmatrix} q_{1,j}\mathbf{q}_j & \cdots & q_{n,j}\mathbf{q}_j \end{bmatrix} \end{aligned}$$

Note that for $1 \leq i \leq n$, column i of $\boldsymbol{Q}^\top$ is given by row i of $\boldsymbol{Q}$, and hence is given by

$$\boldsymbol{Q}_{\cdot i}^T := \begin{bmatrix} q_{i,1} \\ \vdots \\ q_{i,p+1} \end{bmatrix}$$

Now if we sum the projections we get

$$\begin{aligned} \sum_{j=1}^{p+1} \mathbf{q}_j \mathbf{q}_j^\top &= \sum_{k=1}^{p+1} \begin{bmatrix} q_{1,j}\mathbf{q}_j & \cdots & q_{n,j}\mathbf{q}_j \end{bmatrix} \\ &= \begin{bmatrix} \sum_{j=1}^{p+1} q_{1,j}\mathbf{q}_j & \cdots & \sum_{j=1}^{p+1} q_{n,j}\mathbf{q}_j \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{Q} \begin{bmatrix} q_{1,1} \\ \vdots \\ q_{1,p+1} \end{bmatrix} & \cdots & \boldsymbol{Q} \begin{bmatrix} q_{n,1} \\ \vdots \\ q_{n,p+1} \end{bmatrix} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{Q}\boldsymbol{Q}_{\cdot 1}^\top & \cdots & \boldsymbol{Q}\boldsymbol{Q}_{\cdot p+1}^\top \end{bmatrix} \\ &= \boldsymbol{Q} \begin{bmatrix} \boldsymbol{Q}_{\cdot 1}^\top & \cdots & \boldsymbol{Q}_{\cdot p+1}^\top \end{bmatrix} \\ &= \boldsymbol{Q}\boldsymbol{Q}^\top \end{aligned}$$

□

(p) [easy] Explain why adding a new column to $\boldsymbol{X}$ results in no change to SST.

The SST is the SSE of the null model, and the null model does not take into account any of the features; it only looks at the responses. Adding more columns to $\boldsymbol{X}$ amounts

to adding more features (increasing p) and not changing the responses otherwise (n remains fixed). Put another way, the SST is given by

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2.$$

Since adding new columns does not add any responses, we still have the same list of y_i values, and hence the SST stays the same.

(q) [harder] Prove that adding a new column to $\boldsymbol{X}$ results in SSR increasing.

Recall SSR is defined by

$$\begin{aligned} SSR &:= \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 \\ &= \|\hat{\boldsymbol{y}} - \bar{y}\mathbf{1}_n\|^2 \end{aligned}$$

Suppose $\boldsymbol{X}$ has $p+1$ columns. Let $\boldsymbol{X} = \boldsymbol{QR}$ be the QR-decomposition of $\boldsymbol{X}$, and let $\boldsymbol{b}_Q$ be the least squares estimate obtained by projecting onto the column space of $\boldsymbol{Q}$ (which is exactly the column space of $\boldsymbol{X}$). We can assume that $\mathbf{q}_0$, the first column of $\boldsymbol{Q}$, is a scalar multiple of $\mathbf{1}_n$, the first column of $\boldsymbol{X}$. This is reasonable because we can obtain $\boldsymbol{Q}$ by applying Gram Schmidt to $\boldsymbol{X}$, and since the first column of $\boldsymbol{X}$ is $\mathbf{1}_n$, it follows that $\mathbf{q}_0 = \mathbf{1}_n / \|\mathbf{1}_n\|$. Then $\hat{\boldsymbol{y}} = \boldsymbol{Q}b_Q$. Since $\boldsymbol{H} = \boldsymbol{QQ}^\top$, we have

$$\begin{aligned} SSR &:= \|\hat{\boldsymbol{y}} - \bar{y}\mathbf{1}_n\|^2 \\ &= \|\boldsymbol{Hy} - \bar{y}\mathbf{1}_n\|^2 \\ &= \|\boldsymbol{QQ}^\top \boldsymbol{y} - \bar{y}\mathbf{1}_n\|^2 \\ &= \left\| \sum_{j=0}^{p} (\mathbf{q}_j^\top \boldsymbol{y})\mathbf{q}_j - \bar{y}\mathbf{1}_n \right\|^2 \\ &= \left\| (\mathbf{q}_0^\top \boldsymbol{y})\mathbf{q}_0 - \bar{y}\mathbf{1}_n) + \sum_{j=1}^{p} (\mathbf{q}_j^\top \boldsymbol{y})\mathbf{q}_j \right\|^2 \\ &= \left\| \frac{\mathbf{1}_n^\top \boldsymbol{y}}{\|\mathbf{1}_n\|} \frac{\mathbf{1}_n}{\|\mathbf{1}_n\|} - \bar{y}\mathbf{1}_n \right\|^2 + \sum_{j=1}^{p} \|(\mathbf{q}_j^\top \boldsymbol{y})\mathbf{q}_j\|^2 \qquad \text{(by orthogonality)} \\ &= \left\| \underbrace{\frac{n\bar{y}}{\sqrt{n}} \frac{\mathbf{1}_n}{\sqrt{n}} - \bar{y}\mathbf{1}_n}_{\mathbf{0}_n} \right\|^2 + \sum_{j=1}^{p} |(\mathbf{q}_j^\top \boldsymbol{y})|^2 \|\mathbf{q}_j\|^2 \\ &= \sum_{j=1}^{p} |(\mathbf{q}_j^\top \boldsymbol{y})|^2 \qquad \text{(since } \|\mathbf{q}_j\| = 1\text{)} \end{aligned}$$

If we add another column $\boldsymbol{x}_{\cdot p+1}$ to $\boldsymbol{X}$, this entails adding another column $\mathbf{q}_{p+1}$ to $\boldsymbol{Q}$. Therefore, the SSR changes to

$$\begin{aligned} SSR_{\text{new}} &= \sum_{j=1}^{p+1} |(\mathbf{q}_j^\top \boldsymbol{y})|^2 \\ &= SSR + |(\mathbf{q}_{p+1}^\top \boldsymbol{y})|^2 \end{aligned}$$

Thus the SSR increases by $|(\mathbf{q}_{p+1}^\top \boldsymbol{y})|^2$ (since this quantity is non-negative).

(r) [harder] What is overfitting? Use what you learned in this problem to frame your answer.

Overfitting occurs when we use a large number of features, close to n, thereby superficially increasing the SSR associated with our predictions. Put another way, our model ends up fitting the noise in our data, thereby making the model worse.

(s) [easy] Why are the "in-sample" error metrics R^2 and SSE dishonest? Note: I'm leaving out MSE and RMSE as they attempt to be honest by increasing as p increases due to the denominator.

Because we can easily add random data to our design matrix, pretend that they are new features, thereby eventually morphing $\boldsymbol{X}$ into an $n \times n$ matrix of full rank. In the process, the "features" we have added have nothing to do with the true drivers (the z's) of the phenomenon we are modeling. Yet, since $\boldsymbol{X}$ is full rank, the SSE goes to zero and R^2 becomes 1. We get a perfect fit, yet our predictions are worse.

(t) [easy] How can we provide honest error metrics for R^2, SSE? It may help to draw a picture of the procedure.

We can get hold of future data, $\mathbb{D}_*$, and define our R^2 and SSE in terms of the predictions $\hat{\boldsymbol{y}}_*$ on $\mathbb{D}_*$ and the actual responses $\mathbf{y}_*$. Since we don't generally have future data, we partition our data set into two pieces:

$$\begin{aligned} \mathbb{D} &= \mathbb{D}_{train} \cup \mathbb{D}_{test} \\ \emptyset &= \mathbb{D}_{train} \cap \mathbb{D}_{test} \end{aligned}$$

$\mathbb{D}_{test}$ plays the role of $\mathbb{D}_*$ (future data), and it is a proportion of $\mathbb{D}$, whose size is determined by a parameter K:

$$K = \frac{n}{n_{test}} \quad \Longrightarrow \quad \frac{1}{K} = \frac{n_{test}}{n}$$

We build our model using $\mathbb{D}_{train}$, but we define our error metrics in terms of $\mathbb{D}_{test}$. We define the error metrics in terms of $\mathbb{D}_*$:

$$
\begin{aligned}
\mathbf{e}_* &:= \boldsymbol{y}_* - \hat{\boldsymbol{y}}_* \\
oosSSE &:= \mathbf{e}_*^\top \mathbf{e}_* \\
oosMSE &:= \frac{oosSSE}{n_{test}} \\
ooRMSE &:= \sqrt{oosMSE} \\
oosR^2 &:= 1 - \frac{oosSSE}{SST_{test}} \\
SST_* &:= \|\boldsymbol{y}_* - \bar{y_*}\mathbf{1}\|^2
\end{aligned}
$$

(u) [easy] The procedure in the previous question produces highly variable honest error metrics. Can you change the procedure slightly to reduce the variation in the honest error metrics? What is this procedure called and how is it done?

We apply K-fold cross-validation. Starting with data set $\mathbb{D}$, we iteratively perform K splits. In the first split, we let $\frac{1}{K} = \frac{n_{test}}{n}$ of the data be used for $\mathbb{D}_{test}$, while the rest if $\mathbb{D}_{train}$. Once we compute a model using $\mathbb{D}_{train}$, we compute the out-of-sample error metrics using $\mathbb{D}_{test}$. Then, we choose a different $\frac{1}{K}$ portion of the data to use as $\mathbb{D}_{test}$, and compute the errors. We do this a total of K times. The result is a we have K out-of-sample metrics, obtained from the out-of-sample errors $\mathbf{e}_{*1}, \ldots, \mathbf{e}_{*k}$. We can then define:

$$
S_{oosRMSE} := \sqrt{\frac{1}{K-1}\sum_{\ell=1}^{K}(oosRMSE_\ell - \overline{oosRMSE})^2}
$$

which is a measure of standard deviation.

Problem 3

These are some questions related to validation.

(a) [easy] Assume you are doing one train-test split where you build the model on the training set and validate on the test set. What does the constant K control? And what is its tradeoff?

The constant K controls the proportion of the original data set $\mathbb{D}$ that is used as test data. That is, $\frac{1}{K}$ of the observations (row vectors) in $\mathbb{D}$ make up $\mathbb{D}_{test}$; the remaining vectors become $\mathbb{D}_{train}$. The smaller the K, the more data we are using for $\mathbb{D}_{test}$, so our mode will be subject to higher out-of-sample error, but it will be less variable (we can trust it more). The larger the K, the less data we use for $\mathbb{D}_{test}$, and hence the more data in $\mathbb{D}_{train}$, resulting in less out-of-sample error on average but with more variability (harder to trust).

(b) [easy] What problem does K-fold CV try to solve?

K-fold CV tries to address the issue of stability of the error metrics. The out-of-sample error metrics become more reliable, and we can better trust them to estimate the performance of the model.

(c) [difficult] [MA] Theoretically, how does K-fold CV solve this problem? The Internet is your friend.