MATH 342W / 650.4 Spring 2024 Homework #3

Laasya Indrakanti

Monday 18th March, 2024

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 and we will discuss in class. Do your best.

Weather is hard to model the atmosphere especially when there are many uncertainties than can affect it. We aren't able to use past data to make precise predictions that can be useful. Forecasters try to model it using living models to capture the nuances, but it results in limitations.

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

People would rather expect that there's rain and find out that there isn't than to experience rain when they don't expect it, so weathermen exaggerate the chance of rain. To get honest forecasts, you should look for information from forecasters who are transparent about uncertainty.

(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.

Earthquakes don't have consistent indicators for early warnings. It's harder to make predictive models since earthquakes are rare and inconsistent, so there isn't enough data to work with. It's also difficult to predict because they can be triggered by unpredictable factors such as human activity.

(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 lock picking solution because the solutions from the locks provided can't be generalized to other locks.

(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?

This means that four parameters can be used to create a simple model but more parameters add complexity to the model. This tells me that adding more features can be useful, but it is important to be careful with doing so as not to overfit the model.

(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.

Unemployment is affected by various unpredictable factors, such as those relating to human activity, economic policies, or natural disasters. It is often modeled with factors such as GDP or inflation, but these factors have lagged effects so it may not accurately indicate unemployment. There are many other factors that would influence unemployment on an individual level such as demographics, making unemployment difficult to predict.

(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.

I agree because if you don't have any idea, you could end up predicting with random predictors that have nothing to do with the outcome and overfit the model but you might not realize this and therefore make bad predictions.

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 $[\boldsymbol{H}] = p + 1$

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

$$\boldsymbol{HX} = \boldsymbol{X}\underbrace{(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{X}}_{\boldsymbol{I}} = \boldsymbol{XI} = \boldsymbol{X}$$

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

When projecting $\boldsymbol{X}$ onto colsp $[\boldsymbol{X}]$, you just get the original matrix $\boldsymbol{X}$.

(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 $[X']$.

$$\boldsymbol{H} = \boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T$$

Let $\mathbf{S} = [\vec{u_1}|\vec{u_2}|\ldots|\vec{u_{p+1}}]$ where $\mathbf{S}$ is full rank

$$\vec{u_j} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{ which selects a column from } X$$

$$[\vec{\boldsymbol{X}}_{\cdot 1}|\vec{\boldsymbol{X}}_{\cdot 2}|\vec{\boldsymbol{X}}_{\cdot 3}] \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [\vec{\boldsymbol{X}}_{\cdot 2}|\vec{\boldsymbol{X}}_{\cdot 1}|\vec{\boldsymbol{X}}_{\cdot 3}]$$

$$\boldsymbol{X} := \boldsymbol{X}\mathbf{S}$$

$$\boldsymbol{H}' = (\boldsymbol{X}\mathbf{S})((\boldsymbol{X}\mathbf{S})^\top(\boldsymbol{X}\mathbf{S}))^{-1}(\boldsymbol{X}\mathbf{S})^\top = \boldsymbol{X}\mathbf{S}(\mathbf{S}^\top\boldsymbol{X}^\top\boldsymbol{X}\mathbf{S})^{-1}\mathbf{S}^\top\boldsymbol{X}^\top$$

$$= \boldsymbol{X}\mathbf{S}\mathbf{S}^{-1}(\boldsymbol{X}^\top\boldsymbol{X})^{-1}(\mathbf{S}^\top)^{-1}\mathbf{S}^\top\boldsymbol{X}^\top = \boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T = \boldsymbol{H}$$

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

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

Let $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ where $\boldsymbol{A}$ is symmetric and idempotent

$$\boldsymbol{A} = \boldsymbol{A}^\top \text{ and } \boldsymbol{A} = \boldsymbol{A}^2 \implies \boldsymbol{A}^\top\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A} = \boldsymbol{A}$$

Want to prove: $(\vec{y} - \boldsymbol{A}\vec{y})^\top(\boldsymbol{A}\vec{x}) = 0 \ \forall \vec{x}$ because the dot product between the residual vector $(\vec{y} - \boldsymbol{A}\vec{y})$ and any vector in the subspace $(\boldsymbol{A}\vec{x})$ being equal to zero means that the two vectors are orthogonal to each other, proving that A is an orthogonal projection matrix.

$$(\vec{y} - \boldsymbol{A}\vec{y})^\top(\boldsymbol{A}\vec{x}) = (\vec{y}^\top - \vec{y}^\top\boldsymbol{A}^\top)(\boldsymbol{A}\vec{x}) = \vec{y}^\top\boldsymbol{A}\vec{x} - \vec{y}^\top\boldsymbol{A}^\top\boldsymbol{A}\vec{x} = \vec{y}^\top\boldsymbol{A}\vec{x} - \vec{y}^\top\boldsymbol{A}\vec{x} = 0 \quad \blacksquare$$

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

$$I_n = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix} \qquad I_n^\top = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix}$$

$$I_n I_n = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

Since I_n is both symmetric ($I_n = I_n^\top$) and idempotent ($I_n = I_n I_n$), I_n is an orthogonal projection matrix $\forall n$

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

I_n projects onto $\mathbb{R}^n$

(i) [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 model has $p+1$ degrees of freedom. This represents the model's ability to fit the data better as p increases, however this may eventually lead to overfitting.

(j) [easy] If you are orthogonally projecting the vector $\boldsymbol{y}$ onto the column space of X which is of rank $p+1$, derive the formula for $\text{Proj}_{\text{colsp}[X]}[\boldsymbol{y}]$. Is this the same as in OLS?

$\text{Proj}_{\text{colsp}[X]}[\boldsymbol{y}] = \boldsymbol{H}\boldsymbol{y} = \hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\beta} = \boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$

This is the same as in OLS.

(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 $\boldsymbol{0}_n$ than the first $\boldsymbol{e}$? That is, wouldn't this yield a better model on iteration #2? Yes/no and explain.

$\boldsymbol{e}_{new} = \boldsymbol{e}$

Iterating more times wouldn't improve the model because one iteration extracts all the information necessary since (according to econ382) the OLS is the best linear unbiased estimator (BLUE). Also, $\boldsymbol{H}$ is idempotent so multiple iterations makes no difference.

(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.

$$\boldsymbol{Q}^\top\boldsymbol{Q} = \boldsymbol{I} \implies \boldsymbol{Q}^\top\underbrace{\boldsymbol{Q}\boldsymbol{Q}^{-1}}_{\boldsymbol{I}} = \underbrace{\boldsymbol{I}\boldsymbol{Q}^{-1}}_{\boldsymbol{Q}^{-1}} \implies \boldsymbol{Q}^\top = \boldsymbol{Q}^{-1}$$

(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.

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

(n) [difficult] [MA] This problem is independent of the others. Let H be an orthogonal projection matrix. Prove that $\text{rank}\left[\boldsymbol{H}\right] = \text{tr}\left[\boldsymbol{H}\right]$. 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 $\boldsymbol{Q}$.

$$\text{Proj}_{\text{colsp}[Q]}\left[\vec{a}\right] = \boldsymbol{Q}\boldsymbol{Q}^\top = \boldsymbol{H}$$

Let $\boldsymbol{Q} = [\vec{q}_{\cdot 1}, \vec{q}_{\cdot 2}, \ldots, \vec{q}_{\cdot k}]$

$$\text{Proj}_{\text{colsp}[Q]}\left[\vec{a}\right] = \boldsymbol{Q}(\boldsymbol{Q}^\top \boldsymbol{Q})^{-1}\boldsymbol{Q}^\top \vec{a} = \boldsymbol{Q}\boldsymbol{Q}^\top \vec{a}$$

$$\boldsymbol{Q}\boldsymbol{Q}^\top \vec{a} = [\vec{q}_{\cdot 1}, \vec{q}_{\cdot 2}, \ldots, \vec{q}_{\cdot k}]\begin{bmatrix} \vec{q}_{\cdot 1}^\top \\ \vec{q}_{\cdot 2}^\top \\ \vdots \\ \vec{q}_{\cdot k}^\top \end{bmatrix}\vec{a} = \vec{q}_{\cdot 1}(\vec{q}_{\cdot 1}^\top \vec{a}) + \vec{q}_{\cdot 2}(\vec{q}_{\cdot 2}^\top \vec{a}) + \ldots + \vec{q}_{\cdot k}(\vec{q}_{\cdot k}^\top \vec{a})$$

$$= \textstyle\sum_{j=1}^{k} \vec{q}_{\cdot j}(\vec{q}_{\cdot j}^\top \vec{a})$$

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

SST remains constant because it is only a function of the y vector.

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

$$\text{SSR} = \textstyle\sum_{j=1}^{p} ||\text{Proj}_{\vec{q}_{\cdot j}}\left[\vec{y}\right]||$$

$$\boldsymbol{X}_* = [\boldsymbol{X}|\vec{x}_{\cdot *}]$$

$$\boldsymbol{Q}_* = [\boldsymbol{Q}|\vec{q}_*]$$

$$\text{SSR}_* = \text{SSR} + \underbrace{||\text{Proj}_{\vec{q}_*}\left[\vec{y}\right]||^2}_{\geq 0} \implies \text{SSR}_* \geq \text{SSR}$$

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

Overfitting is what happens when we have added $\vec{x}_*$'s irrelevant to the proximal causes and our SSR and R^2 seem to fit the data more closely when in reality the model is getting worse if not staying the same.

(s) [easy] Why are "in-sample" error metrics (e.g. R^2, SSE, s_e) dishonest? Note: I'm leaving out RMSE as RMSE attempts to be honest by increasing as p increases due to the denominator. I've chosen to use standard error of the residuals as the error metric of choice going forward.

They are dishonest because they are manipulable by adding features which may not always be relevant. If junk features are added, R^2 will increase and SSE will decrease, but the model is not actually improving in performance.

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

We can use $\mathbb{D}_{\text{future}}$ to calculate the out of sample metrics. Since we don't have $\mathbb{D}_{\text{future}}$, we assume stationarity and split our $\mathbb{D}$ into $\mathbb{D}_{\text{train}}$ and $\mathbb{D}_{\text{test}}$ and treat the $\mathbb{D}_{\text{test}}$ as our $\mathbb{D}_{\text{future}}$.

(u) [easy] The procedure in (t) 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 can change the procedure to reduce variation in the metrics by splitting the data k times. This is called k-fold cross-validation.

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?

K controls the proportion of $\mathbb{D}$ that gets put into $\mathbb{D}_{\text{test}}$. If $\mathbb{D}_{\text{train}}$ is too small, estimation error increases, but if $\mathbb{D}_{\text{test}}$is too small, we can't really trust the validation (we wouldn't have enough predictions to make a solid decision). There is a tradeoff between bias and variance of the error metrics.

(b) [harder] Assume you are doing one train-test split where you build the model on the training set and validate on the test set. If n was very large so that there would be trivial misspecification error even when using $K = 2$, would there be any benefit at all to increasing K if your objective was to estimate generalization error? Explain.

For n large, there would be marginal benefit to increasing K, not enough to justify doing so. $K = 2$ means that $n_{test} = 0.5n$ but if n is large, the training set would have enough data to use for predictions.

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

K-fold CV attempts to reduce variance among oos metrics, especially if the test set is too small.

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

It allows for more training data since we can make more predictions even with a smaller test set and it averages the results so it results in more accurate metrics.