MATH 342W / 650.4 Spring 2024 Homework #3

Professor Adam Kapelner

Tuesday 19th 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.

The problem with weather predictions within the machine learning framework can be described in terms of data collection, model selection, and the inherent unpredictability of weather phenomena. The important and non generic notation we would use for phenomenon would be:

- t: Time step at which a prediction is made.
- Y or y: The target variable we aim to predict, such as future temperature, precipitation, or storm intensity.
- f: The true underlying function that maps $\mathbf{X}$ to Y, which in the context of weather prediction is unknown.
- $\mathcal{D}$: The dataset containing historical weather observations $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ used to train the model g.

The challenge with weather prediction using this is how f is described and used. We can rely on this function properly as the phenomenon of the atmosphere itself is very abrupt and hard to make predictions on.

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

Over predicting to show caution towards the rain is a big reason why weathermen "lie." This is because its safer to predict that it will rain even if its not extremely clear that it will rain. To get honest forecasts you should go to the National Weather Services (NWS). The author emphasizes their commitment to providing unbiased and accurate forecasts.

(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.

There are multiple problems with earthquake predictions, one of them being that false alarms and missed predictions are both a factor. This enables two different ways to get an error. Making predictions with the abundance of seismic data and formulating it into a way that can accurately tell earthquakes is exremely complex. The notations could represent variables and functions such as time (t), forecasting functions (f, g), errors (ϵ, δ), datasets (D), hypotheses (H), and various statistical measures and parameters. Adding these parameters is the part which seems complex because you wouldnt know how much to weight each one especially because it might increase false alarms and on the other hand also predict falsely.

(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?

Silver's nonsense predictor is the overly specific plan that the criminals derived to pick a lock based on its color. This is "nonsense" because it is not generalizable and the color of the lock is irrelevant to the lock combination.

(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?

What von Neumann meant is the potential for complex models with many parameters to fit the data very closely. This would also capture random noise as if it were meaningful signal. This is because adding more parameters to a model increases its ability to conform more precisely to the uniqueness of the dataset it's trained on

(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.

As Silver says, the problem with predicting unemployment stems from the unpredictability of the economic system which differs from predicting weather and earthquakes. Overfitting using historical context is a huge issue where many times especially in stock trading, the economic models may overfit past data such as finding patterns that appear to be to be likely but are instead just coincidental.

(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.

Silver encourages a more holistic approach, where data points are viewed as manifestations of deeper, underlying principles. By grounding our predictions in theories about how things function, we enable a more nuanced and potentially accurate forecast. I do agree, having theories support predictions can be referred to repeatedly. Even if your predictions are correct without any theory backing it, it could become hard to replicate it. When basing them off theories, it could ensure more repeatability.

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 $[\boldsymbol{H}]$?

Given that H is the orthogonal projection onto colsp[X] where X is an n $\times(p+1)$ matrix with all columns linearly independent, the rank of H is $p+1$.

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

Given$H = X(X^TX)^{-1}X^T$, then $HX = X(X^TX)^{-1}X^TX$ Since X^TX is invertible, we simplify to HX = X

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

The result HX = X conceptually means that projecting the columns of X onto the space spanned by themselves results in X. This shows that X is fully contained within its own colsp. By decomposing X into an orthogonal matrix Q and an upper triangular matrix R, shows the direct relationship between orthogonal projections and least squares linear regression.

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

To show that the projection matrix that projects onto colsp[X] is the same as the one that projects onto colsp[X'], where X' is X with its columns in reverse order, we consider the orthogonal projection matrix for X given by $H = X(X^TX)^{-1}X^T.ForX'$, the projection matrix would be $H' = X'(X'^TX')^{-1}X'^T$. Since the columns of X and X' span the same subspace (they contain the same vectors in a reverse order), the column spaces of X and X' are identical.

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

MA

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

MA

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

The identity matrix I_n acts as an orthogonal projection matrix because it projects any vector v in n-dimensional space onto itself which shows idempotence. This makes $I_n \in$ orthogonal projection matrix, where the projection space is the entire n-dimensional space itself.

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

The subspace that the matrix I_n projects onto the entire n-dimensional space itself.

(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?

In least squares linear regression using a design matrix X with rank $p+1$, the degrees of freedom in the resulting model are $p+1$. This represents the number of independent parameters in the model which are p predictor variables and 1 intercept.

(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?

The formula for the orthogonal projection of a vector y onto the column space of X ($\text{Proj}_{\text{colsp}[X]}[y]$) is given by $H = X(X^TX)^{-1}X^T$, so the projection of y is $Hy = X(X^TX)^{-1}X^Ty$. This is the same as (OLS) solution for the estimated outcomes $\hat{y}$, where $\hat{y} = X\beta$ and $\beta = (X^TX)^{-1}X^Ty$. So: $Hy = \hat{y}$ 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.

Because e is orthogonal to the column space of X, after we regress again in linear least squares regression we would not get residuals closer to 0_n in the second iteration.

(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.

We can derive for the orthonormal matrix Q the property that $Q^T = Q^{-1}$ by involking the definition of orthonormal itself.

(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.

To prove that the least squares projection matrix $H = X(X^TX)^{-1}X^T = QQ^T$, where Q is the orthogonal matrix obtained from the QR decomposition of X, we can QR decomposition itself: $X = QR$, R is a matrix.

Since Q is orthogonal, $Q^T Q = I$. Thus,

$$H = X(X^T X)^{-1} X^T = QR(R^T Q^T QR)^{-1} R^T Q^T$$

Because $R^T Q^T QR = R^T R$, and $Q^T Q = I$, this simplifies to:

$$H = QR(R^T R)^{-1} R^T Q^T$$

This simplifies further to:

$$H = QQ^T$$

This at the end shows that $\boldsymbol{H} = QQ^T$

(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.

MA

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

The orthogonal projection onto the column space of Q can be shown to be equivalent to the sum of projections onto each of its columns. Let Q be composed of columns $q_1, q_2, ..., q_n$. The projection of a vector y onto colsp$[Q]$ is $QQ^T y$.

Since $Q = [q_1 \; q_2 \; ... \; q_n]$, QQ^T can be viewed as a sum of outer products of the columns of Q:

$$QQ^T = q_1 q_1^T + q_2 q_2^T + ... + q_n q_n^T$$

Each term $q_i q_i^T$ represents the projection onto the line spanned by q_i. Therefore, the projection onto colsp$[Q]$ is the sum of the projections onto its column vectors.

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

SSR is determined by variance in y, which is not subject to changes because SST measures total variability in y which is independent of the model's predictors.

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

SSR refects the variance in y. Adding a new column to X can increase the SSR since it provides more information on the variance in y. with more explanatory variables, the model fits the data more closely.

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

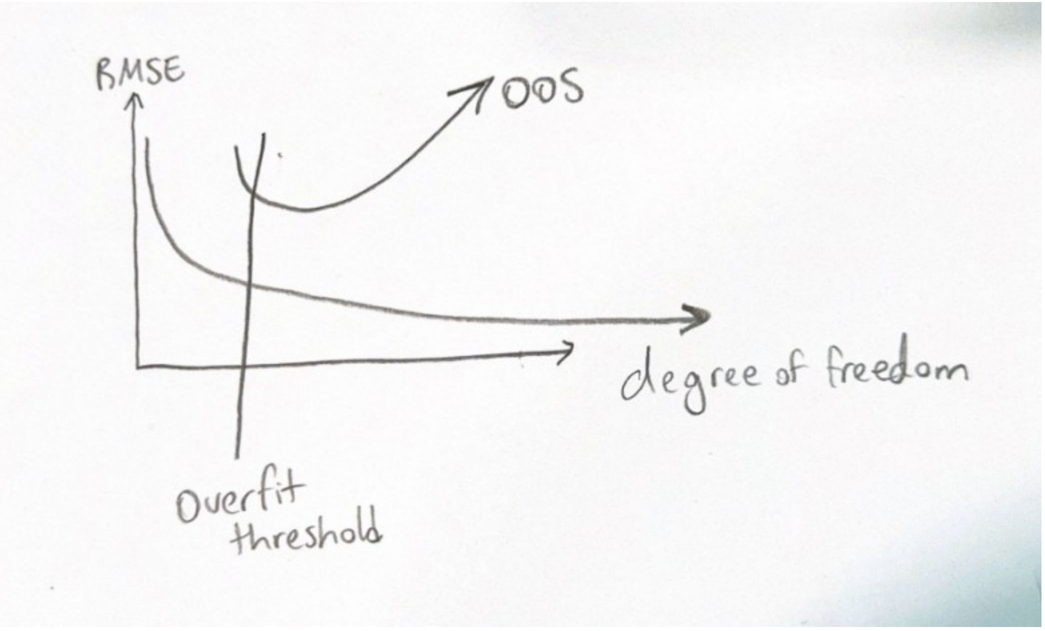
Overfitting occurs when a statistical model or machine learning algorithm captures the noise of the data instead of the underlying relationship. From what we've discussed, adding more parameters to a model (like additional columns to X in linear regression) can increase the sum of squares due to regression (SSR) and decrease the residual sum of squares (SSE), making the model appear to fit the data better. On the other hand, if the new parameters do not showcase patterns and instead inherit random coincidences, the model would then overfit with information and data that should not be used by the predictors but still are, meaning the accuracy of the model would be worse.

(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.

"In-sample" error metrics like R^2, SSE, and the se are considered "dishonest" because they evaluate the model's performance on the same data it was trained on. As a model becomes more complex (for example, by increasing p, the number of predictors), it tends to fit the training data more closely, potentially capturing "noise" as if it were a "signal". This can lead to overly optimistic performance estimates. The "in-sample" would give a higher accuracy rating when in actuality the "in-sample" metrics should show that the accuracy of the model is worsening.

(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.

As shown in class we created new metrics that were honest and showed a massive improvement from our in-sample metrics. Here is a graph that compares performances:



(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?

To reduce the variation in honest error metrics that we got in (t), we can use k-fold cross-validation. We would be randomly dividing the entire dataset into k equally (or nearly equally) sized segments or folds. Then, the model is trained on $k-1$ folds and tested on the remaining fold. This process is repeated k times, each time with a different fold used as the test set. This is a good way to split the dataset and also show accuracy in the model.

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 K constant controls the number of folds that will take place in the dataset. The tradeoff is the accuracy of the model itself being accurate or not. This is because if the test is too large and train is too small, the model may not have enough data to train off of Vice-Versa.

(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.

With a larger K, each training set is more representative of the overall dataset, leading to a more accurate and stable estimate of the model's performance on unseen data. Although there are many benefits to increasing K, if my objective was to estimate generalization error, then there wouldn't be much benefit as the results wouldn't be as beneficial as they were earlier.

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

If the available dataset is not large enough, K-fold tries to solve this and does so well by giving a solution with the dataset you already have.

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