MATH 390.4 / 650.2 Spring 2018 Homework #4t

Joseph Peltroche

Monday 7th May, 2018

Problem 1

These are questions about Silver's book, chapters ... 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. and also we now have $f_{pr}, h^*_{pr}, g_{pr}, p_{th}$, etc from probabilistic classification as well as different types of validation schemes).

(a) [easy] What algorithm that we studied in class is PECOTA most similar to?

Solution. PECOTA is most similar to KNN, i.e. K- nearest neighbors. □

(b) [easy] Is baseball performance as a function of age a linear model? Discuss.

Solution. Baseball performance as a function of age is in a constant state of flux, but resembles somewhat an inverted parabola. Players tend to peak a certain age then gradually regress to a downward slope. □

(c) [harder] How can baseball scouts do better than a prediction system like PECOTA?

Solution. Baseball scouts have access to the same information as PECOTA but can also rely on their expertise and human judgement that PECOTA could never do. This allows for the opportunity to reject any player that might be producing favorable stats but does truly not have much skill. □

(d) [harder] Why hasn't anyone (at the time of the writing of Silver's book) taken advantage of Pitch f/x data to predict future success?

Solution. The introduction of new technology described in Pitch f/x may change its emphasis toward the things that are even harder to quantify. This will make our $\mathcal{D}$ grow, and allow our dimensions to grow, i.e., $p \uparrow$.However, this causes a greater blur between stats and scotuing, and while it is a very powerful and promising tool, it takes great skill to use it the Pitch f/x data in a smart way or figure out how to fuse quantitative and qualitative equations of player performance. This could potentially worsen a predictability ($\hat{\boldsymbol{y}}$) if not used correctly. □

(e) [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.

Solution. The main issue with predicting weather is the unpredictablity. The weather system obeys dynamically properties. Thus, it is non-linear, abiding to an exponential rather than additive relationships, and the behavior of the system at one point in time influences the behavior in the future. The difficulty lies in the attempting to minimize the error when small changes to initial conditions can spur huge and unexpected divergence in outcomes. These initial conditions, our parameters, would have to be so precise in order to not create huge amounts of error. □

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

Solution. Weatherman lie about the chance of rain to please the public. Should it be forecasted to not rain on a certain day and it actually does rain, then the general public will be upset. Should the reverse happen, then the public will interpret this as a fortitious outcome. If one desires honest forecasts, one should attain information directly from spots like the Hurricane Center. □

(g) [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.

Solution. The problems with predicting earthquakes are rooted in the underdeveloped theory and the noisy data. Contrary to weather prediction, there is much less theoretical understanding of the earth's crust than the earth's atmosphere. Therefore the human judgment that can be added to the model becomes useless to optimize the predictability, i.e. improve $\hat{y}$. Additionally, there is not a strong $\mathcal{D}$ to obtain, since the predictors are all noisy (cannot be directly measured). This produces a lot of noise, e, for predictions and undermines future theory for how the earthquakes are really formed. □

(h) [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?

Solution. The nonsense predictor he describes in the model found in page 163 is color lock. □

(i) [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?

Solution. He means that with additional parameter, there is more that can be done on a system. For a data scientist, this means that predicitablity improves upon a bigger parameter space. □

(j) [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.

Solution. The problem with unemployment predictions is the know as correlation without causation. What this means is when predicting a response variable such as unemployment, certain predictors might have a statistical relationship between one another but are actually independent. When looking at the macroeconomy and interpreting different statistical relationships between variables x_p, y, there may be no causation in reality. To conclude there is causation when there actually isn't would produce much more noise e and push us further from the signal y. □

(k) [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.

Solution. I would agree, so that when dealing with complex systems like earth quakes, weather, or the economy, there will be theoretical background one can rely to straighten out any outliers the statistical models may have overlooked or missed. Having a pundit's expertise increases the chances of greater predictabliity. □

Problem 2

This question is about validation for the supervised learning problem with one fixed $\mathbb{D}$.

(a) [easy] For one fixed $\mathcal{H}$ and $\mathcal{A}$ (i.e. one model), write below the steps to do a simple validation and include the final step which is shipping the final g.

Solution. The steps for model selevtion and validation are as follows:

(i) Split the training set into $\mathcal{D}_{train}, \mathcal{D}_{select}, \mathcal{D}_{test}$

(ii) create a g that is equal to $\mathcal{A}(\mathcal{H}, \mathcal{D}_{train})$

(iii) Find out of sample error $oose = error(y_{select}, g(X_{select}))$.

(iv) Find the out of sample error of the model by using $\mathcal{D}_{test}$, i.e., $oose = error(y_{test}, g(X_{test}))$.

(v) Go through steps 2-5 on $\mathcal{D}$ to produce g_{final} and compare the previous g chosen from the steps 2-4.

□

(b) [easy] For one fixed $\mathcal{H}$ and $\mathcal{A}$ (i.e. one model), write below the steps to do a K-fold cross validation and include the final step which is shipping the final g.

Solution. The protocol for K-fold cross validation is as follows:

(i) fit $g_K = \mathcal{A}(\mathcal{H}, \mathcal{D}_{train,K})$.

(ii) find the predictions produced by g_K on X_{test} . $\vec{\hat{y}}_K = g_K(X_{test,K})$

(iii) Repeat the above two steps for 1,..., K folds

(iv) Concatenate vertically $\vec{\hat{y}}_{cv} = \begin{bmatrix} \vec{\hat{y}}_1 \\ \vdots \\ \vec{\hat{y}}_K \end{bmatrix}$

(v) compare the out of sample error $oose = error(y, \vec{\hat{y}}_{cv})$

□

(c) [harder] For one fixed $\mathcal{H}$ and $\mathcal{A}$ (i.e. one model), write below the steps to do a bootstrap validation and include the final step which is shipping the final g.

Solution. (a) fit $g_{ki,k0} = \mathcal{A}(\mathcal{H}, \mathcal{D}_{train,ki,k0})$

(b) Compute $\vec{\hat{\boldsymbol{y}}}_{ki,k0} = g_{ki,k0}(\mathcal{D}_{select,ki,k0})$

(c) Repeat steps 1-2 for all inner folds $ki \in \{1, \ldots, 5\}$

(d) Concatenate

$$\vec{\hat{\boldsymbol{y}}}_{k0} = \begin{bmatrix} \vec{\hat{\boldsymbol{y}}}_{1,k0} \\ \vdots \\ \vec{\hat{\boldsymbol{y}}}_{5,k0} \end{bmatrix}$$

(e) Find the *oose*

(f) Repeat steps 1-5 for $k_0 \in \{1, \ldots, 5\}$

(g) set

$$\vec{\hat{\boldsymbol{y}}} = \begin{bmatrix} \hat{\boldsymbol{y}}_1 \\ \vdots \\ \hat{\boldsymbol{y}}_5 \end{bmatrix}$$

(h) Estimate $oose = error(\vec{\hat{\boldsymbol{y}}}, \vec{y})$

(i) Repeat steps 1-6 to build final model g without $\mathcal{D}_{test}$.

□

(d) [harder] For one fixed $\mathcal{H}_1, \ldots \mathcal{H}_M$ and $\mathcal{A}$ (i.e. M different models), write below the steps to do a simple validation and include the final step which is shipping the final g.

Solution. The steps for model selevtion and validation are as follows:

(i) Split the training set into $\mathcal{D}_{train}, \mathcal{D}_{select}, \mathcal{D}_{test}$

(ii) create a g_j that is equal to $\mathcal{A}_j(\mathcal{H}_j, \mathcal{D}_{train})$

(iii) Compute the jth out of sample error using y_{select} and $\mathcal{D}_{select}$, i.e., $oose_j = error(y_{select}, g_j(X_{select})$

(iv) Repeat steps 2-3 for all $j = \{1, 2, \ldots, M\}$ models.

(v) Find the j that corresponds to the smallest out of sample error $j^* = argminoose_1, \ldots, oose_M$.

(vi) Find the out of sample error of that j^* model by using $\mathcal{D}_{test}$, i.e., $oose_{j^*} = error(y_{test}, g_{j^*}(X_{test}))$.

(vii) Go through steps 2-5 on $\mathcal{D}$ to produce g_{final} and compare the previous g chosen from the master alogirthm (steps 2-5).

□

(e) [difficult] For one fixed $\mathcal{H}_1, \ldots \mathcal{H}_M$ and $\mathcal{A}$ (i.e. M different models), write below the steps to do a K-fold cross validation and include the final step which is shipping the final g. This is not an easy problem! There are a lot of steps and a lot to keep track of...

Solution. (a) fit $g_{j,ki,k0} = \mathcal{A}_j(\mathcal{H}_j, \mathcal{D}_{train,ki,k0})$

(b) Compute $\vec{\hat{\boldsymbol{y}}}_{j,ki,k0} = g_{j,ki,k0}(\mathcal{D}_{select,ki,k0}$

(c) Repeat steps 1-2 for all models $j \in \{1, \ldots, M\}$

(d) Repeat steps 1-2 for all inner folds $ki \in \{1, \ldots, 5\}$

(e) Concatenate

$$\vec{\hat{\boldsymbol{y}}}_{j,k0} = \begin{bmatrix} \vec{\hat{\boldsymbol{y}}}_{j,1,k0} \\ \vdots \\ \vec{\hat{\boldsymbol{y}}}_{j,5,k0} \end{bmatrix}$$

(f) Select best model $æ^*_{k0} = argmin\{oose_{1,k0}, \ldots, osse_{M,k0}\}$

(g) Repeat steps 1-6 for $k_0 \in \{1, \ldots, 5\}$

(h) set

$$\vec{\hat{\boldsymbol{y}}} = \begin{bmatrix} \hat{\boldsymbol{y}}_{j^*,1} \\ \vdots \\ \hat{\boldsymbol{y}}_{j^*,5} \end{bmatrix}$$

(i) Estimate $oose = error(\vec{\hat{\boldsymbol{y}}}, \vec{y})$

(j) Repeat steps 1-6 to build final model g without $\mathcal{D}$.

□

Problem 3

This question is about ridge regression — an alternative to OLS.

(a) [harder] Imagine we are in the "Luis situation" where we have $\boldsymbol{X}$ with dimension $n \times (p+1)$ but $p+1 > n$ and we still want to do OLS. Why would the OLS solution we found previously break down in this case?

Solution. The dimensions of our $\boldsymbol{X}$ have changed to the point where $n < p+1$. The rank of $\boldsymbol{X}$ is now less than $p+1$. Therefore theres exists a non-zero vector $\vec{u}$ in $\mathcal{R}^{p+1}$ that produces the zero vector upon multiplying it with $\boldsymbol{X}$. This indirectly means that $X^T X$ is not at full rank

$$X^T X \vec{u} = X^T 0_{p+1} = 0_{p+1}$$

Therefore, we cannot find the inverse to $X^T X$ and cannot use the OLS expression for b. □

(b) [harder] We will embark now to provide a solution for this case. The solution will also give nice results for other situations besides the Luis situation as well. First, assume λ is a positive constant and demonstrate that the expression $\lambda \left|\left|\boldsymbol{w}\right|\right|^2 = \boldsymbol{w}^\top (\lambda \boldsymbol{I}) \boldsymbol{w}$ i.e. it can be expressed as a quadratic form where $\lambda \boldsymbol{I}$ is the determining matrix. We will call this term $\lambda \left|\left|\boldsymbol{w}\right|\right|^2$ the "ridge penalty".

Solution. Starting off with $\boldsymbol{w}^\top (\lambda \boldsymbol{I}) \boldsymbol{w}$, we will show that this is equivalent to $\lambda \left|\left|w\right|\right|^2$

$$\boldsymbol{w}^\top_{1\times(p+1)} (\lambda \boldsymbol{I})_{(p+1)\times(p+1)} \boldsymbol{w}_{(p+1)\times 1} = \begin{bmatrix} \omega_0 & \omega_1 & \dots & \omega_p \end{bmatrix} \begin{bmatrix} \lambda & 0 & \dots & 0 \\ 0 & \lambda & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & \lambda \end{bmatrix} \begin{bmatrix} \omega_0 \\ \omega_1 \\ \vdots \\ \omega_p \end{bmatrix}$$

$$= \begin{bmatrix} \lambda\omega_0 & \lambda\omega_1 & \dots & \lambda\omega_p \end{bmatrix} \begin{bmatrix} \omega_0 \\ \omega_1 \\ \vdots \\ \omega_p \end{bmatrix} = \lambda \boldsymbol{w}^\top \boldsymbol{w} = \lambda ||\boldsymbol{w}||^2$$

□

(c) [easy] Write the $\mathcal{H}$ for OLS below where there parameter is the $\boldsymbol{w}$ vector. $\boldsymbol{w} \in ?$

Solution.

$$\mathcal{H} = \{\omega_0 + \omega_1 x_1 + \ldots + \omega_p x_p | \boldsymbol{w} \in \mathcal{R}^{p+1}\}$$

□

(d) [easy] Write the error objective function that OLS minimizes using vectors, then expand the terms similar to the previous homework assignment.

Solution.

$$\begin{aligned} SSE = \sum_i^n (y_i - \hat{y}_i)^2 &= (\vec{y} - \vec{\hat{y}})^T (\vec{y} - \vec{\hat{y}}) && (\vec{v}^T \vec{v} = ||\vec{v}||^2) \\ &= (\vec{y}^T - \vec{\hat{y}}^T)(\vec{y} - \vec{\hat{y}}) \\ &= \vec{y}^T \vec{y} - \vec{y}^T \vec{\hat{y}} - \vec{\hat{y}}^T \vec{y} + \vec{\hat{y}}^T \vec{\hat{y}} \\ &= \vec{y}^T \vec{y} - 2\vec{\hat{y}}^T \vec{y} + \vec{\hat{y}}^T \vec{\hat{y}} && (a^T b = b^T a) \\ &= \vec{y}^T \vec{y} - 2(X\vec{w})^T \vec{y} + (X\vec{w})^T (X\vec{w}) \\ &= \vec{y}^T \vec{y} - 2\vec{w}^T X^T \vec{y} + \vec{w}^T X^T X \vec{w} && ((ab)^T = b^T a^T) \end{aligned}$$

□

(e) [easy] Now add the ridge penalty $\lambda ||\boldsymbol{w}||^2$ to the expanded form you just found and write it below. We will term this two-part error function the "ridge objective".

Solution.

$$\vec{y}^T \vec{y} - 2\vec{w}^T X^T \vec{y} + \vec{w}^T X^T X \vec{w} + \lambda ||\boldsymbol{w}||^2$$

□

(f) [easy] Note that the ridge objective looks a bit like the hinge loss we spoke about when we were learning about support vector machines. There are two pieces of this error function in counterbalance. When this is minimized, describe conceptually what is going on.

Solution. When the error function is minimized, the $\boldsymbol{w}$ vector is opitiziming the function such that the ridge penalty is at the smallest value while causing the other piece to not "explode", i.e., dramatically increase in value. Rather, both pieces will be the smallest value such that the other will not dramatically increase to produce the minimum value of the cost function. □

(g) [harder] Now, the ridge penalty term as a quadratic form can be combined with the last term in the least squares error from OLS. Do this, then use the rules of vector derivatives we learned to take $d/d\boldsymbol{w}$ and write the answer below.

Solution.

$$SSE = \vec{y}^T\vec{y} - 2\boldsymbol{w}^T X^T\vec{y} + \boldsymbol{w}^T X^T X\boldsymbol{w} + \boldsymbol{w}^\top(\lambda \boldsymbol{I})\boldsymbol{w}$$

$$\frac{d}{d\boldsymbol{w}}SSE = \vec{0}_{p+1} - 2X^\top\vec{y} + 2X^\top X\boldsymbol{w} + 2\lambda\boldsymbol{I}\boldsymbol{w}$$

□

(h) [easy] Now set that derivative equal to zero. What matrix needs to be invertible to solve?

Solution.

$$\frac{d}{d\boldsymbol{w}}SSE = \vec{0}_{p+1} - 2X^\top\vec{y} + 2X^\top X\boldsymbol{w} + 2\lambda\boldsymbol{I}\boldsymbol{w} = \vec{0}_{p+1}$$

$$\Rightarrow X^\top X\boldsymbol{w} + \lambda\boldsymbol{I}\boldsymbol{w} = X^\top\vec{y}$$

$$(X^\top X + \lambda\boldsymbol{I})\boldsymbol{w} = X^\top\vec{y}$$

$$\boldsymbol{w} = (X^\top X + \lambda\boldsymbol{I})^{-1}X^T\vec{y}$$

The matrix $(X^\top X + \lambda\boldsymbol{I})$ must be an invertible matrix in order to solve for $\boldsymbol{w}$. □

(i) [difficult] There's a theorem that says *positive definite* matrices are invertible. A matrix is said to be positive definite if every quadratic form is positive for all vectors i.e. if $\forall \boldsymbol{z} \neq \boldsymbol{0}\ \ \boldsymbol{z}^\top A\boldsymbol{z} > 0$ then A is positive definite. Prove this matrix from the previous question is positive definite.

(j) [easy] Now that it's positive definite (and thus invertible), solve for the $\boldsymbol{w}$ that is the argmin of the ridge objective, call it $\boldsymbol{b}_{ridge}$. Note that this is called the "ridge estimator" and computing it is called "ridge regression" and it was invented by Hoerl and Kennard in 1970.

Solution.

$$\frac{d}{d\boldsymbol{w}}SSE = \vec{0}_{p+1} - 2X^\top\vec{y} + 2X^\top X\boldsymbol{w} + 2\lambda\boldsymbol{I}\boldsymbol{w} = \vec{0}_{p+1}$$

$$\Rightarrow X^\top X\boldsymbol{w} + \lambda\boldsymbol{I}\boldsymbol{w} = X^\top\vec{y}$$

$$(X^\top X + \lambda\boldsymbol{I})\boldsymbol{w} = X^\top\vec{y}$$

$$\boldsymbol{b}_{ridge} = (X^\top X + \lambda\boldsymbol{I})^{-1}X^T\vec{y}$$

□

(k) [easy] Did we just figure out a way out of Luis's situation? Explain.

Solution. By incorporating a ridge penalty term for the $\boldsymbol{X}$ matrices that have the dimensions $n \times (p+1)$ where $p+1 > n$ in our vector form of our cost function, we were able to find an analytic solution for the $\boldsymbol{b}$ vector. The ridge penalty term fixed the matrix that we needed to be invertible to become invertible (proven in part (i)), and proceeded analogously to the LS method. □

(l) [harder] It turns out in the Luis situation, many of the values of the entries of $\boldsymbol{b}_{ridge}$ are close to 0. Why should that be? Can you explain now conceptually how ridge regression works?

Solution. Conceptually, ridge regression works in an analgous way to the LS solution. The introduction of a penalty term merely changes the the inverse of the matrix we previously found the inverse for in LS. The strength of the penalty term depends on the value of λ and will influence the elements of the coefficient vector $\boldsymbol{b}_{ridge}$. The elements of the $\boldsymbol{b}_{ridge}$ vector are near zero due to the selection of the λ hyperparameter. In order to optimize the expression, a certain range of λ is allowed and that range will cause the entries of the $\boldsymbol{b}_{ridge}$ to be near zero. □

(m) [easy] Find $\hat{\boldsymbol{y}}$ as a function of $\boldsymbol{y}$ using $\boldsymbol{b}_{ridge}$. Is $\hat{\boldsymbol{y}}$ an orthogonal projection of $\boldsymbol{y}$ onto the column space of $\boldsymbol{X}$?

Solution.

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{b}_{ridge}$$

The $\hat{\boldsymbol{y}}$ is an orthogonal projection of $\boldsymbol{y}$ onto the column space of $\boldsymbol{X}_{ridge}$, not $\boldsymbol{X}$. This would include information by the $\boldsymbol{X}$ matrix and the information from the ridge penalty to create a new matrix $\boldsymbol{X}_{ridge}$. □

(n) [E.C.] Show that this $\hat{\boldsymbol{y}}$ is an orthogonal projection of $\boldsymbol{y}$ onto the column space of some matrix $\boldsymbol{X}_{ridge}$ (which is not $\boldsymbol{X}$!) and explain how to construct $\boldsymbol{X}_{ridge}$ on a separate page.

(o) [easy] Is the $\mathcal{H}$ for OLS the same as the $\mathcal{H}$ for ridge regression? Yes/no.
Is the $\mathcal{A}$ for OLS the same as the $\mathcal{A}$ for ridge regression? Yes/no.

Solution. The candidate space $\mathcal{H}$ is the same since we are still looking for parameters of $\boldsymbol{w} \in \mathcal{R}$ but the algorithim has changed since we are dealing with a ridge penalty term in our analytics and now have a ridge objective. OLS did not consider this term, although the methods are similar. □

(p) [harder] What is a good way to pick the value of λ, the hyperparameter of the $\mathcal{A}$ = ridge?

Solution. We would ideally want to pick a hyperparameter λ that is not too small or not to big since there is a trade off between both pieces of the ridge objective. Numerical methods work effectively to attain a range of hyperparameters that meet this quota. Some methods include quadratic programming, sub-gradient descent, and coordinate descent. □

(q) [easy] In classification via $\mathcal{A}$ = support vector machines with hinge loss, how should we pick the value of λ? Hint: same as previous question!

Solution. Similarly to the previous question, we should rely on numerical methods to obtain a λ that would work best to minimze our expression as shown in class. □

(r) [E.C.] Besides the Luis situation, in what other situations will ridge regression save the day?

(s) [difficult] The ridge penalty is beautiful because you were able to take the derivative and get an analytical solution. Consider the following algorithm:

$$\boldsymbol{b}_{lasso} = \underset{\boldsymbol{w} \in \mathbb{R}^{p+1}}{\arg\min} \left\{ (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) + \lambda ||\boldsymbol{w}||^1 \right\}$$

This penalty is called the "lasso penalty" and it is different from the ridge penalty in that it is not the norm of $\boldsymbol{w}$ squared but just the norm of $\boldsymbol{w}$. It turns out this algorithm (even though it has no closed form analytic solution and must be solved numerically a la the SVM) is very useful! In "lasso regression" the values of $\boldsymbol{b}_{lasso}$ are not shrunk *towards* 0 they are harshly punished *directly to* 0! How do you think lasso regression would be useful in data science? Feel free to look at the Internet and write a few sentences below.

Solution. Despite the "lasso regression" depending on numerics, it can come in handy and become useful in data science when dealing with reductions. This regression will push certain parameters to 0, and make the model much more simpler and interpretable. The ridge regression, however, will include all terms and reduce some paramters near 0 but never exactly to 0 (include useless information), complicating the model even more. □

(t) [easy] Is the $\mathcal{H}$ for OLS the same as the $\mathcal{H}$ for lasso regression? Yes/no.
Is the $\mathcal{A}$ for OLS the same as the $\mathcal{A}$ for lasso regression? Yes/no.

Solution. The candidate space $\mathcal{H}$ is the same as for OLS since we are still considering $\boldsymbol{w} \in \mathcal{R}^\top$ but the alogirthim is different now that we are considering an additional term , i.e., the "lasso penalty". □

Problem 4

These are questions about non-parametric regression.

(a) [easy] In problem 1, we talked about schemes to validate algorithms which tried M different prespecified models. Where did these models come from?

Solution. These models came from a "model space", a space with different specified models produced by the many $\mathcal{H}$ and alogirthims $\mathcal{A}$. □

(b) [harder] What is the weakness in using M pre-specified models?

Solution. The weakness is having specified models that are not actually the closest possible model to the data. □

(c) [difficult] Explain the steps clearly in forward stepwise linear regression.

Solution. In forward stepwise linear regression, □

(d) [difficult] Explain the steps clearly in *backwards* stepwise linear regression.

(e) [harder] What is the weakness(es) in this stepwise procedure?

Solution. The weaknesses that occur with stepwise procedure is that it is rough on the numerics. Solving this numerically will take a lot of time and computational power to run. Another weakness is the susceptibility to over fitting and under fitting. This depends on the intervals and really runs into a problem when dealing with distributions that seem similar to a step function. □

(f) [easy] Define "non-parametric regression". What problem(s) does it solve? What are its goals? Discuss.

Solution. The regression "non-parametric regression" is such that the candidate space of our model does not take a predetermined form. Rather, the form will be constructed according to the data distribution. It solves the problem of neglecting any possible models that could have been a better fit if we had chosen to used a parametric regression and specificed a candidate space. □

(g) [harder] Provide the steps for the regression tree (the one algorithm we discussed in class) below.

Solution. (i) Begin with alll $\boldsymbol{X}, \vec{y}$

(ii) For every possible split at the current node, divide the data into left: $\boldsymbol{X}_L, \vec{y}_L$ and right $\boldsymbol{X}_R, \vec{y}_R$ and compute their respective SSE: $SSE_L = \sum_{i=1}^{n_L}(y_{Li} - \bar{y}_l)^2$ and $SSE_R = \sum_{i=1}^{n_R}(y_{Ri} - \bar{y}_l)^2$

(iii) Find the split which miniizes the "overall" SSE,SSE_{avg}

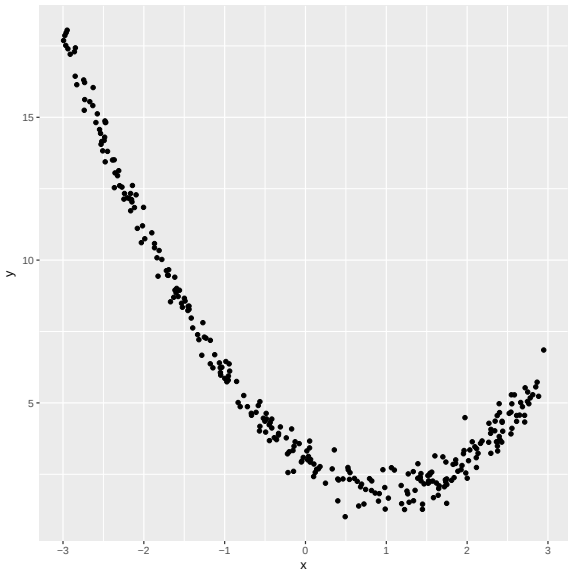
$$SSE_{avg} = \frac{n_L SSE_L + n_R SSE_R}{n_L + n_R}$$

(iv) Create that split and assign $\boldsymbol{X}_L, \vec{y}_L$ and $\boldsymbol{X}_R, \vec{y}_R$ to the daughter nodes

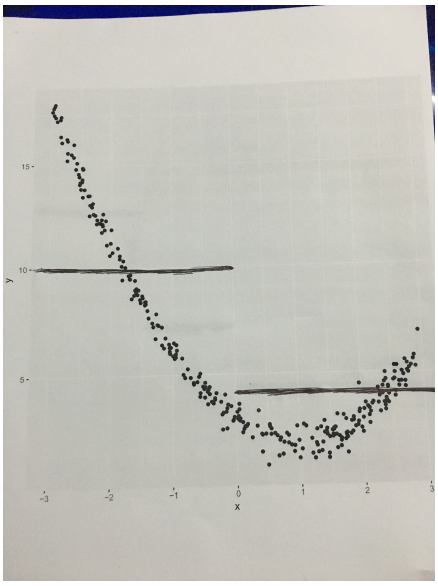
(v) Recurse on steps 2-4 for both daughter nodes until "STOP"

□

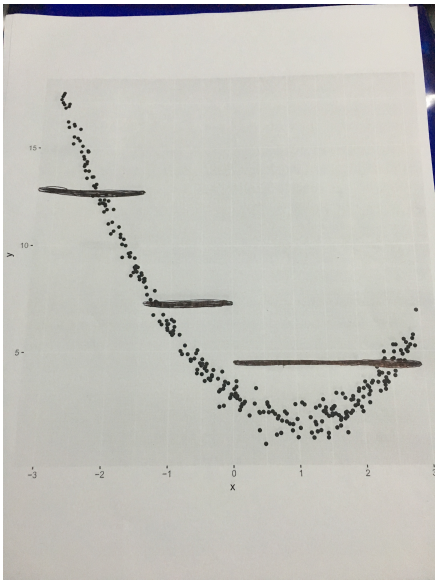
(h) [easy] Consider the following data



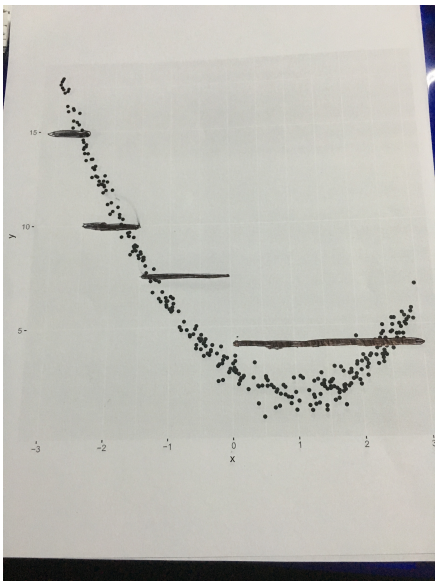
Create a tree with maximum depth 1 (i.e one split at the root node) and plot g above.



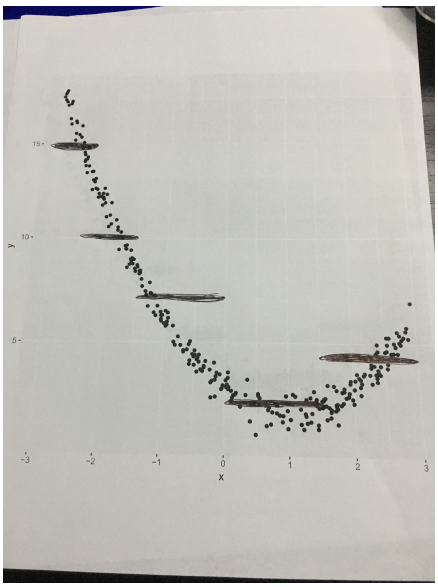
(i) [easy] Now add a second split to the tree and plot g below.



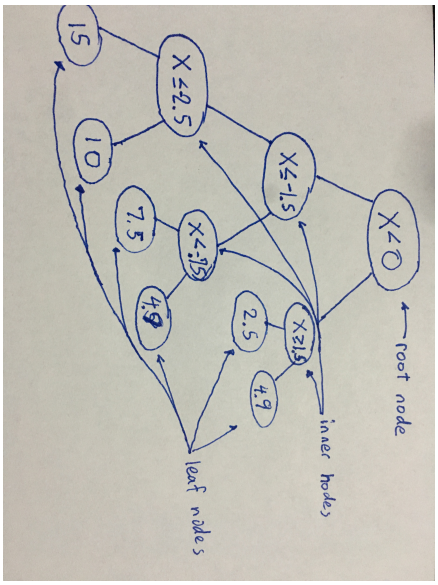
(j) [easy] Now add a third split to the tree and plot g below.



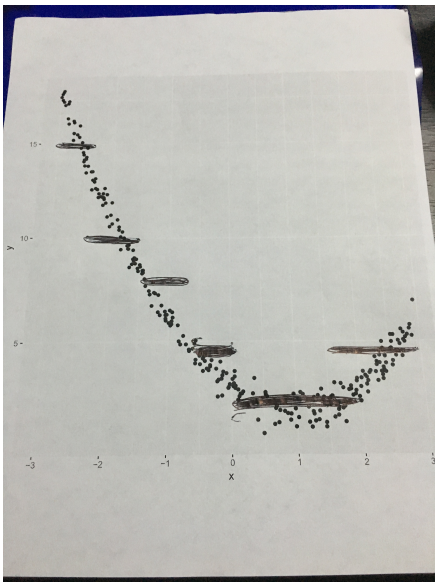
(k) [easy] Now add a fourth split to the tree and plot g below.



(l) [easy] Draw a tree diagram of g below indicating which nodes are the root, inner nodes and leaves. Indicate split rules and leaf values clearly.



(m) [easy] Plot g below for the mature tree with the default $N_0 =$ `nodesize` hyperparameter.



(n) [easy] If $N_0 = 1$, what would likely go wrong?

Solution. If $N_0 = 1$, then the tree will grow to fit a separate parameter for each data point and R^2 will dramatically reach 100% (overfitting). □

(o) [easy] How should you pick the $N_0 =$ `nodesize` hyperparameter in practice?

Solution. The hyperparameter N_0 should be picked via model selection procedure. □

Problem 5

These are questions about classification trees.

(a) [easy] How are classification trees different than regression trees?

Solution. The difference between the two trees is the output space they are attempting to predict on. The regression tree will output a numerical result, while the classification tree is will output into a categorical space. □

(b) [harder] What are the steps in the classification tree algorithm?

Solution. (i) Start with all data

(ii) For every possible split, calculate the "Gini impairing"

$$\text{Gini}_L = \sum_{\ell=1}^{k} \hat{P}_\ell(1-\hat{P}_\ell) \qquad \text{Gini}_R = \sum_{\ell=1}^{k} \hat{P}_\ell(1-\hat{P}_\ell)$$

$$\hat{P}_\ell = \frac{\#y_i \text{ in label } \ell}{\#\text{ of observations in node}}$$

(iii) Find the splits with the lowest neighborhood average

$$\text{Gini}_{Avg} = \frac{n_L Gini_L + n_R Gini_R}{n_L + n_R}$$

(iv) Create a split and portion data in left, right, and daughter nodes.

(v) For the left, set data in step 1, repeat steps 2-5. Do the same for the right until "STOP" :)number of observations is less than N_0.)

(vi) For all leaf nodes, assign $\hat{\boldsymbol{y}} = Mod[\vec{y}_0]$ where $\vec{y}_0$ is the vector of the average of y_i's in the leaf node.

□

Problem 6

These are questions about measuring performance of a classifier.

(a) [easy] What is a confusion table?

Solution. A confusion table is a table that describes the performance of a classification model on a set of test data for which the true values are known. □

Consider the following in-sample confusion table where "> 50K" is the positive class:

```
       y_hats_train
y_train <=50K >50K
  <=50K  3475  262
  >50K    471  792
```

(b) [easy] Calculate the following: n (sample size) = 5000

FP (false positives) = 262

TP (true positives) = 792

FN (false negatives) = 471

TN (true negatives) = 3475

$\#P$ (number positive) = 1263

$\#N$ (number negative) = 3737

$\#PP$ (number predicted positive) = 1054

$\#PN$ (number predicted negative) = 3946

$\#P/n$ (prevalence / marginal rate / base rate) = 0.2526

$(FP + FN)/n$ (misclassification error) = 0.1466

$(TP + TN)/n$ (accuracy) = 0.8534

$TP/\#PP$ (precision) = 0.7514

$TP/\#P$ (recall, sensitivity, true positive rate, TPR) = 0.6271

$2/(\text{recall}^{-1} + \text{precision}^{-1})$ (F1 score) = 0.6836

$FP/\#PP$ (false discovery rate, FDR) = 0.2486

$FP/\#N$ (false positive rate, FPR) = 0.07011

$FN/\#PN$ (false omission rate, FOR) = 0.1194

$FN/\#P$ (false negative rate, FNR) = 0.3729

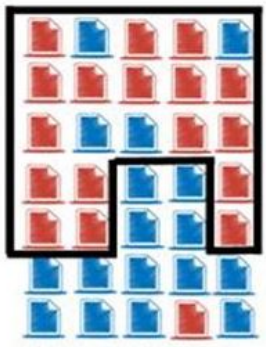
(c) [easy] Why is FPR also called the "false alarm rate"?

Solution. The FPR is also called the “false alarm rate" because it counts the number of false positives, or number of incorrectly predicted 1’s (positive), by the model per the number of true negatives. □

(d) [easy] Why is FNR also called the “miss rate”?

Solution. The FNR is also called the “miss rate" because it calculates the number of false negatives, i.e., the number of times the model predicted 0 when it was actually 1 (or misses), over the actual number of positives (1’s). □

(e) [easy] Below let the red icons be the positive class and the blue icons be the negative class.



The icons included inside the black border are those that have $\hat{y} = 1$. Compute both precision and recall.

Solution.

$$\text{recall} = \frac{TP}{\#P} = \frac{16}{17} \approx 0.9412 \qquad \text{precision} = \frac{TP}{\#PP} = \frac{16}{21} \approx 0.7619$$

□

(f) [harder] There is always a tradeoff of FP vs FN. However, in some situations, you will look at FPR vs. FNR. Describe such a classification scenario. It does not have to be this income amount classification problem, it can be any problem you can think of.

Solution. A classic example would a a fire alarm system. When predicting actual fires, one would want more false positive than false negatives. By increasing the number of positives predicted, then the number of negtives predicted decrease and, thus, the FPR increases as the FNR decreases. The more fire alarms that go off for minor events (not necessarily actual fires), the less of a chance the fire alarm system will predict no fire when there actually is a fire. $\#N$, the true number of negatives, remains invariant, as the model cannot modify this quantity. □

(g) [harder] There is always a tradeoff of FP vs FN. However, in some situations, you will look at FDR vs. FOR. Describe such a classification scenario. It does not have to be this income amount classification problem, it can be any problem you can think of.

Solution. A another classication problem would be determining whether a cancer is malignant (negative) or benign (positive). I would look at the FDR (false discovery rate) and FOR (false omission rate) in order to determine the rate of my negative predictions being false vs the positive predictions being false to minimize the FOR as much as possible (ideally both but practically speaking I can only minimize one). □

(h) [harder] There is always a tradeoff of FP vs FN. However, in some situations, you will look at precision vs. recall. Describe such a classification scenario. It does not have to be this income amount classification problem, it can be any problem you can think of.

Solution. When attempting to make a sale on credit cards through email, it can help to know the precision and recall (if you access to the true values as well). A salesman may have a relativitly high recall rate, selling credit cards to almost everyone who is willing to buy one. However, precision tells a different story. He would probably have to target a massive audience to get those numbers and the precision would consequently be extremely low. □

(i) [harder] There is always a tradeoff of FP vs FN. However, in some situations, you will look only at an overall metric such as accuracy (or $F1$). Describe such a classification scenario. It does not have to be this income amount classification problem, it can be any problem you can think of.

Solution. For the same situation as the previous question, the credit card salesman, evaluating a good salesman requires looking at each salesman $F1$ score. This may be the result of budget cuts and the boss will have to let some people go (we are the boss). Ideally, we would want to keep the best people in our company. This balances out both precision and recall, so that the a salesman that is wasting excessive resources to get considerably well recall scores will score a low $F1$ score. The reverse is also true (high precision, low recall). Therefore, the $F1$ score will distinguish a better salesman from the others. □