MATH 390.4 / 650.2 Spring 2018 Homework #4t

Rebecca Strauss

May 7, 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?
Nearest neighbors

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

It could be, but it would have to be a linear model with many interactions. Also the data is extremely noisy because each player peaks at a different point.

(c) [harder] How can baseball scouts do better than a prediction system like PECOTA?
They can see things like height and weight that computers can't.

(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?

The technology needed to measure x didn't exist yet or was too expensive.

(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.

Weather systems are dynamic. It is impossible to know exactly where your variables are at any given moment because they are constantly moving(notaion). So $f(x)$ is moving too quickly to be measured, this increases our error due to ignorance δ. Also in dynamic systems, our outputs at one stage become our inputs at the next, which makes the error build up exponentially.

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

To create the perception of accuracy for consumers. If the weatherman forecasts a 50% chance of rain, consumers might think he is 'whishy washy', even though that is exactly

what the model predicted. Arbitrarily rounding 10% up or down tricks consumers into thinking the weatherman's predictions are more accurate than they really are. If you want honest forecasts you should go to a nonprofit weather organization like the National Weather Service.

(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.

They can't measure their variables x directly because the activity they are trying to model is occurring deep under the earths surface. Their variables are all expressed in terms of past earthquakes. While climatologist have explicit equations to solve, seismologist don't have an $\mathcal{A}$ to put their data into.

(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?

The color of the locks is the nonsense predictors.

(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?

The more parameters you use in your model, the more you be able to 'manipulate' your model into giving the predictions you desire.

(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.

Unemployment predictions don't account for all the people who are unemployed. The government has a specific definition for unemployment. If you don't fall under the governments specific definition, even if you don't have a job, yo will not be classified as 'unemployed'. So the predictions are not a good indicator of the macroeconomic performance of a country,

(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.

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.

(a) Pick a k, where $1/k$ is the proportion of data saved to be $\mathbb{D}_{test}$

(b) Split $\mathbb{D} = \mathbb{D}_{train} \cup \mathbb{D}_{test}$

(c) Build the model. $g = \boldsymbol{A}(\mathbb{D}_{train}, \mathcal{H})$

(d) Get out-of-sample statistics $\hat{\boldsymbol{y}}_{oos} = g(\boldsymbol{X}_{test})$, $E_{out} = E(\boldsymbol{Y}_{test}, \hat{\boldsymbol{y}}_{oos})$

(e) Build $g_{final} = \mathcal{A}(\mathbb{D}, \mathcal{H})$

(f) Ship error statistics g and g_{final}

(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.

(a) Randomly split $\mathbb{D}$ into k different bins

(b) Repeat for $i = 1..k$

i. Set $\mathbb{D}_{test_i}$ to be bin i and set $\mathbb{D}_{train_i}$ to be everything except for bin i.

ii. Fit $g_i = \mathcal{A}(\mathbb{D}_{train_i}, \mathcal{H})$

iii. Save $\hat{\vec{\boldsymbol{y}}}_i = g_i(\boldsymbol{X}_{test_i})$

(c) Concatenate vertically $\hat{\vec{y}}_{cv} = \begin{bmatrix} \hat{\vec{\boldsymbol{y}}} \\ \vdots \\ \hat{\vec{\boldsymbol{y}}} \end{bmatrix}$

(d) Repeat this processes many times and take the average of all $\hat{\vec{y}}_{cv}$

(e) Compute oos $E_{out} = E(\boldsymbol{Y}, \hat{\vec{y}}_{cv})$

(f) Build and ship $g, g_{final} = \mathcal{A}(\mathbb{D}, \mathcal{H})$ with error statistics

(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.

(a) Randomly sample $\mathbb{D}$ with replacement to create $\mathbb{D}_{train}, \mathbb{D}_{test}$

(b) Build $g = \mathcal{A}(\mathbb{D}_{train}, \mathcal{H})$

(c) Get out-of-sample statistics $\hat{\boldsymbol{y}}_{oos} = g(\boldsymbol{X}_{test})$, $E_{out} = E(\boldsymbol{Y}_{test}, \hat{\boldsymbol{y}}_{oos})$

(d) Build $g_{final} = \mathcal{A}(\mathbb{D}, \mathcal{H})$

(e) Ship error statistics g and g_{final}

(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.

(a) Randomly split $\mathbb{D}$ into $\mathbb{D} = \mathbb{D}_{train} \cup \mathbb{D}_{select} \cup \mathbb{D}_{test}$

(b) For each model $j \in 1..M$

i. Build $g_j = \mathcal{A}(\mathbb{D}_{train}, \mathcal{H}_j)$

ii. Calculate out-of-sample error $E_{out_j} = E(\boldsymbol{Y}_{select, g_j(\boldsymbol{X}_{select})}$

(c) Select best model based on oos statistics $g_j* = argmin\{E_{out_1}, ..E_{out_M}\}$

(d) Compute $E_{out_{j*}} = E(\boldsymbol{Y}_{test}, g_{j*}(X_{test}))$

(e) Build g_{final} with steps (b),(c) using $\mathbb{D}$

(f) Ship g_{j*}, g_{final} and error statistics

(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...

(a) Randomly split $\mathbb{D}$ into k different folds
For each model $j \in 1..M$

i. Fit $g_{ij} = \mathcal{A}(\mathbb{D}_{train_i}, \mathcal{H}_j)$

ii. Compute $\vec{\hat{y}}_i = g_{ji}(\boldsymbol{X}_{test_i})$

iii. Repeat for all folds

(b) Concatanate $\vec{\hat{y}}_j = \begin{bmatrix} \vec{\hat{\boldsymbol{y}}} \\ \vdots \\ \vec{\hat{\boldsymbol{y}}} \end{bmatrix}$

(c) Calculate $E_{out_j} = E(\boldsymbol{Y}, \vec{\hat{y}}_j)($

(d) Repeat for all models $j \in 1...M$

(e) Select best model $g_{j*} = argmin\{E_{out1}, ..E_{out_M}\}$

(f) Calculate $E_out = E(\boldsymbol{Y}, E_{j*})$

(g) $g_final = \mathcal{A}(\mathbb{D}, \mathcal{H})$

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?

The matrix $\boldsymbol{X}$ will not be full rank, so we can't invert to solver for the least squares solution

(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 $\lambda \, ||\boldsymbol{w}||^2 = \boldsymbol{w}^\top (\lambda \boldsymbol{I}) \boldsymbol{w}$ i.e. it can be expressed as a quadratic form where $\lambda \boldsymbol{I}$ is the determining matrix. We will

call this term $\lambda ||\boldsymbol{w}||^2$ the "ridge penalty".

$\lambda ||\boldsymbol{w}||^2 = (\lambda \boldsymbol{I})\boldsymbol{w}^\top \boldsymbol{w} = \boldsymbol{w}^\top (\lambda \boldsymbol{I})\boldsymbol{w}$

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

$\mathcal{H} = \{\boldsymbol{w} \cdot \boldsymbol{X} : \boldsymbol{w} \in \mathbb{R}^{n \times (p+1)}\}$

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

$$\begin{aligned} \sum (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w})^2 &= \\ (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w})^\top (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w}) &= \\ \boldsymbol{Y}^\top \boldsymbol{Y} - \boldsymbol{w}^\top \boldsymbol{X}^\top \boldsymbol{Y} - \boldsymbol{Y}^\top \boldsymbol{X}\boldsymbol{w} + \boldsymbol{w}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{w} &= \\ \boldsymbol{Y}^\top \boldsymbol{Y} - 2\boldsymbol{w}^\top \boldsymbol{X}^\top \boldsymbol{Y} + \boldsymbol{w}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{w} & \end{aligned}$$

(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".

$\boldsymbol{Y}^\top \boldsymbol{Y} - 2\boldsymbol{w}^\top \boldsymbol{X}^\top \boldsymbol{Y} + \boldsymbol{w}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{w} + \boldsymbol{w}^\top (\lambda \boldsymbol{I})\boldsymbol{w}$

(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.

(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.

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{w}}(\boldsymbol{Y}^\top \boldsymbol{Y}) - 2\frac{\partial}{\partial \boldsymbol{w}}(\boldsymbol{w}^\top \boldsymbol{X}^\top \boldsymbol{Y}) + \frac{\partial}{\partial \boldsymbol{w}}(\boldsymbol{w}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{w}) &= \\ 0 - 2\boldsymbol{X}^\top \boldsymbol{Y} + 2\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{w} + 2\lambda \boldsymbol{I}\boldsymbol{w} &= \\ -\boldsymbol{X}^\top \boldsymbol{Y} + \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{w} + \lambda \boldsymbol{I}\boldsymbol{w} & \end{aligned}$$

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

$$\begin{aligned} -\boldsymbol{X}^\top \boldsymbol{Y} + \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{w} + \lambda \boldsymbol{I}\boldsymbol{w} = 0 \\ \lambda \boldsymbol{I}\boldsymbol{w} = \boldsymbol{X}^\top \boldsymbol{Y} - \boldsymbol{X}^\top \boldsymbol{w}^\top \boldsymbol{X} \\ \lambda \boldsymbol{I}\boldsymbol{w} + \boldsymbol{X}^\top \boldsymbol{w}^\top \boldsymbol{X} = \boldsymbol{X}^\top \boldsymbol{Y} \\ \boldsymbol{w}(\lambda \boldsymbol{I} + \boldsymbol{X}^\top \boldsymbol{X}) = \boldsymbol{X}^\top \boldsymbol{Y} \end{aligned}$$

The matrix $\lambda \boldsymbol{I} + \boldsymbol{X}^\top \boldsymbol{X}$ must be invertible

(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 $\forall \boldsymbol{z} \neq \boldsymbol{0} \quad \boldsymbol{z}^\top A \boldsymbol{z} > 0$ then A is positive definite. Prove this matrix from the previous question is positive definite.

(a) $A = \lambda \boldsymbol{I} + \boldsymbol{X}^\top \boldsymbol{X}$

(b) $A > 0$ because

i. $\boldsymbol{X}^\top \boldsymbol{X} > 0$

ii. λ is defined to be a positive constant

(c) $\boldsymbol{z}^\top \boldsymbol{z} > 0$

(d) So $\boldsymbol{z}^\top A \boldsymbol{z} > 0 \forall \boldsymbol{z} \neq 0$

Hence A 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.

$b_{ridge} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top \boldsymbol{Y}$

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

Before we couldn't take the inverse of $X^\top \boldsymbol{X}$ because it's not full rank. With this new definite positive definition, we can take the inverse without need the matric to be full rank.

(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?

(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 $\boldsymbol{X}$?

$\hat{\boldsymbol{y}} = \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top \boldsymbol{Y}$
$\hat{\boldsymbol{y}}$ is not a projection of $\boldsymbol{y}$ onto the column space of $\boldsymbol{X}$

(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.

$\mathcal{H}$ is the same $\mathcal{A}$ is not

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

Graph different values of λ and decide visually/graphically which fit is best.

(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!

Try different values of λ and see which classifies the most data correctly.

(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.

Lasso regression is useful when you want your variables picked and normalized by one method.

(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.

$\mathcal{H}$ is the same $\mathcal{A}$ is not

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?

From an $\mathcal{A}, \mathcal{H}$ defined by us.

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

It doesn't allow for an expressive model.

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

(a) Use the Null model created from $\mathbb{D}_{train}$ as your baseline.

(b) Create a huge $\mathcal{H}$ by using a huge amount of derived predictors and calculate the fits for each one $g_j = \mathcal{A}(H_j, \mathbb{D}_{train})$

(c) For each fit, calclate $oos_j = E(\boldsymbol{Y}_{select}, g_j(\boldsymbol{X}_s elect))$

(d) Iteratively add the 'best' predictors to the null model. 'Best' is defined by what you're looking for. Here best means lowest *oos*.

(e) Computer $oos_{j*} = E(\boldsymbol{Y}_{test}, g_{j*}(\boldsymbol{X}_{test}$

(f) Build model on full $\mathbb{D}$ and ship with everything above

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

(a) Start with a model created from as many predictors as possible in $\mathbb{D}_{train}$

(b) Iteratively delete the 'worst' predictors from the model. Worst here means highest error rate calculated using $\mathbb{D}_{select}$

(c) Delete predictors until you reach the null model + one predictor, or until you are satisfied with your results

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

(a) We still need to specify an intelligent set of predictors. How can we know which combination is the best?

(b) Model is still linear, thus not as expressive as it could be.

(c) Computation time

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

In "non-parametric regressions, we don't prespecify the model space $\mathcal{H}$. $\mathcal{H}$ adjusts itself according to the data. This solves the problem of us, the human, having to define an intelligent set of predictors by ourself. The goal of non-parametric regression is to construct a model from the data alone. The data defines how complex the model is.

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

(a) Begin with all data and pick an N_0 for a stopping point

(b) A every split of data, $<X_l, \vec{Y_l}>$ and $<X_r, \vec{Y_r}>$ and Calculate $SSE_l = \sum(y_l - \bar{y}_l)^2$ and $SSE_r = \sum(y_r - \bar{y}_r)^2$

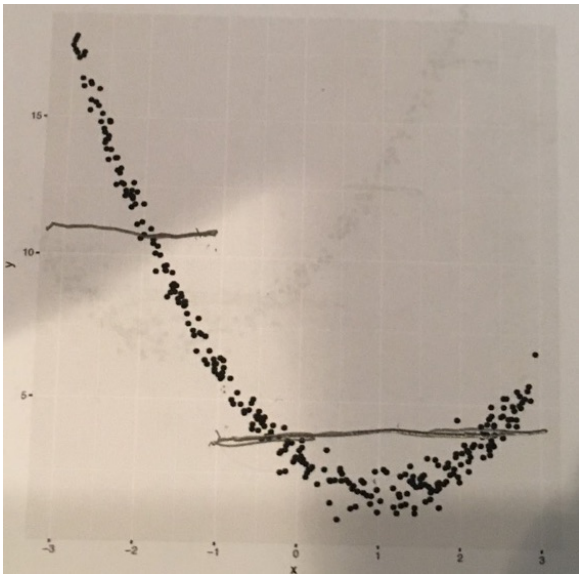
(c) Find the split with the lowest total SSE, $SSE_{tot} = SSE_l + SSE_r$

(d) Create the split. Now $<X_l, \vec{Y_l}>$ and $<X_r, \vec{Y_r}>$ becomes the data in step 1

(e) Recurse until node has $\leq N_0$ data points.

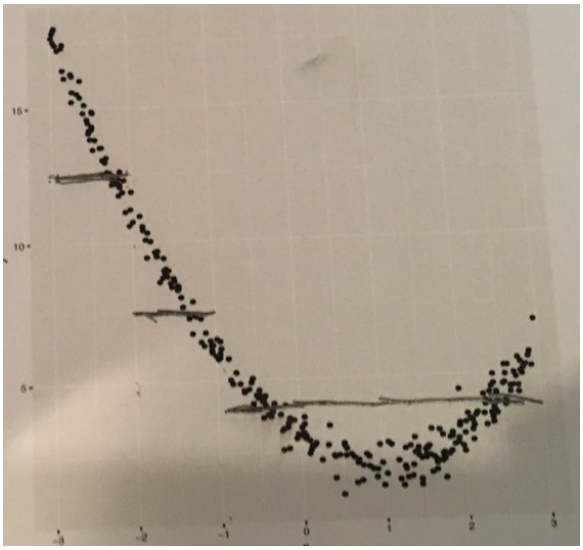
(f) For all leaf nodes, assign $\hat{y} = \bar{y}_0$, where $\bar{y}_0$ is the sample average of all y's in that node.

(h) [easy] Consider the following data

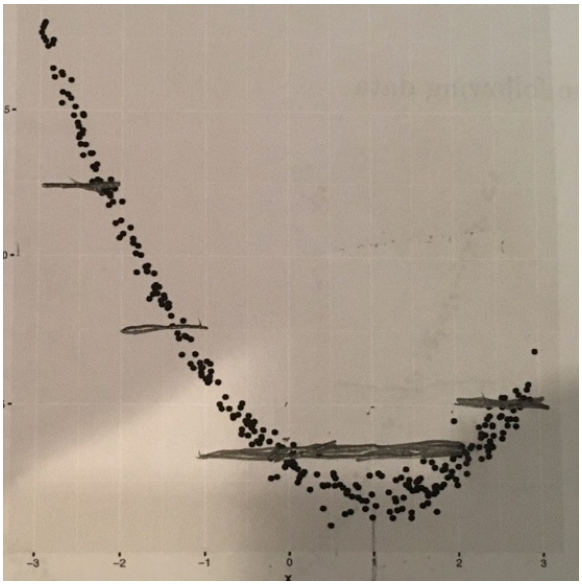


Create a tree with maximum depth 1 (i.e one split at the root node) and plot g above.

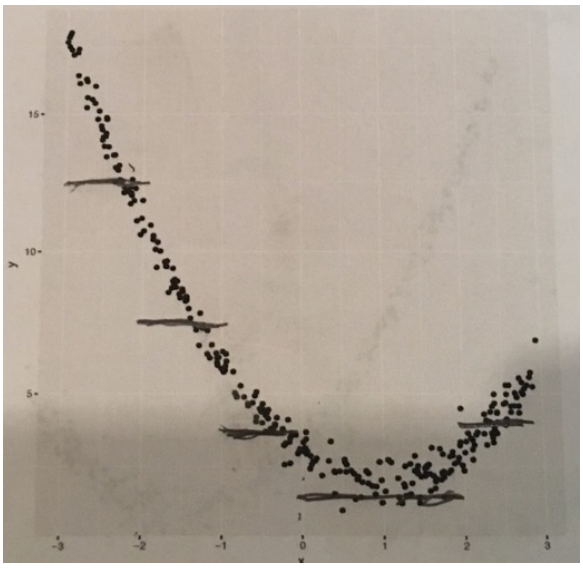
(i) [easy] Now add a second split to the tree and plot g below.



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

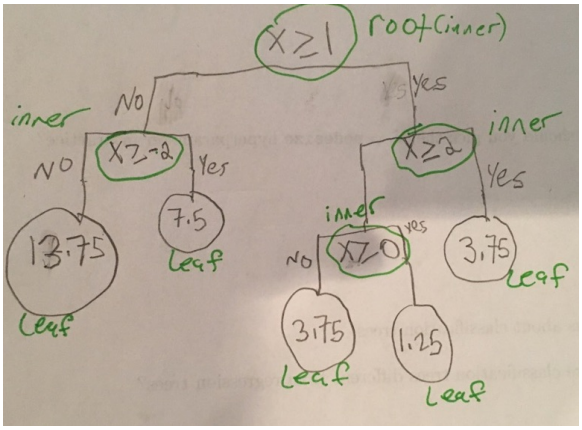


(k) [easy] Now add a fourth split to the tree and plot g below.

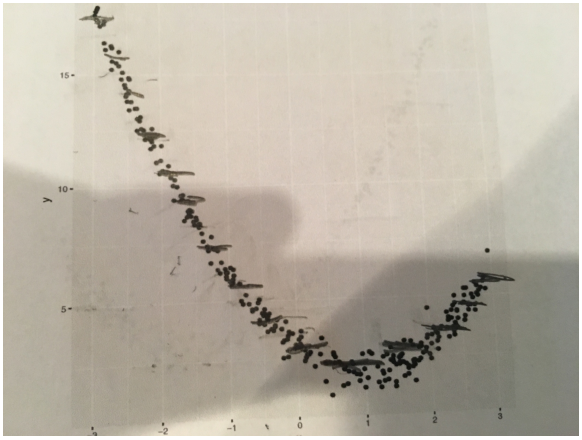


(l) [easy] Draw a tree diagram of g below indicating which nodes are the root, inner nodes and leaves. Indicate split rules and leaf values clearly.

-



(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?

Over fitting

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

Model selection process.

Problem 5

These are questions about classification trees.

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

Classification assigns a label, $y \in \{1 \cdots k\}$. Regression trees assign a real number $Y \in \mathbb{R}$

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

(a) Begin with all training data, choose hyperparameter N_0(usually $N_0 = 1$

(b) For every possible split, calculate the Gini Impurity

$$G_L = \sum_{l=1}^{k} \hat{p}_l(1 - \hat{p}_l) \qquad G_R = \sum_{r=1}^{k} \hat{p}_r(1 - \hat{p}_r)$$

where each $\hat{p}$ equals the amount of y_i in that label(l/r) divided by n(number observations in node)

$$\hat{p}_l = \frac{y_{Ltot}}{n_L} \qquad \hat{p}_r = \frac{y_{Rtot}}{n_R}$$

(c) Find and create the split with the lowest weighted Gini metric

$$G_{avg} = \frac{n_L G_L + n_R G_R}{n_L + n_R}$$

(d) Sort data into left and right daughter nodes correctly

(e) Repeat steps b-d for both daughter nodes until node has less than N_0 observations in it.

(f) For all leaf nodes, assign $\hat{y} = Mode[\vec{y_0}]$ where $\vec{y_0}$ is the average of the 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?

A way to visualize misclassification error.

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) = 475

$\#P$ (number positive) $= FN + TP = 1263$

$\#N$ (number negative) $= FP + TN = 3737$

$\#PP$ (number predicted positive) $= FP + TP = 1054$

$\#PN$ (number predicted negative) $= TN + FN = 3946$

$\#P/n$ (prevalence / marginal rate / base rate) $= 1263/5000 = .25$

$(FP + FN)/n$ (misclassification error) $= (262 + 471)/5000 = .15$

$(TP + TN)/n$ (accuracy) $= (792 + 3475)/5000 = .85$

$TP/\#PP$ (precision) $= 792/1054 = .75$

$TP/\#P$ (recall, sensitivity, true positive rate, TPR) $= 792/1263 = .63$

$2/(\text{recall}^{-1} + \text{precision}^{-1})$ (F1 score) $= \frac{2}{\frac{1}{.63} + \frac{1}{.75}} = .68$

$FP/\#PP$ (false discovery rate, FDR) $= 262/1054 = .25$

$FP/\#N$ (false positive rate, FPR) $= 262/3737 = .07$

$FN/\#PN$ (false omission rate, FOR) $= 471/3946 = .12$

$FN/\#P$ (false negative rate, FNR) $= 471/1263 = .37$

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

It represents how often a 'no' gets misclassified as a 'yes'.

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

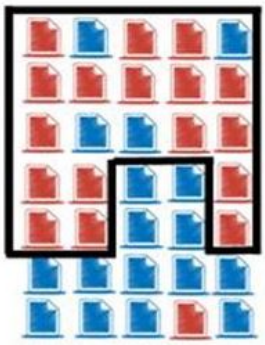
It represents how often a 'yes' is misclassified as 'no'.

(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.

Precision: $TP/\#PP = 4/21$ Recall: $TP/\#P = 4/17$

(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.



When classifying tumors with the breast cancer data. Here, FPR would be what proportion of benign tumors get classified as cancerous. FNR would be what proportion of cancerous tumors get classified as benign.

(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.

In the breast cancer data set, FDR would be what proportion of tumors that we predicted cancerous, were actually benign. FOR would represent wat proportion of tumors we predicted benign were actually cancerous.

(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.

In the breast cancer data set, precision would represent what proportion of tumors we predict cancerous, were actually cancerous. Recall would represent what proportion of all cancerous tumors did we label as cancerous.

(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.

in the breast cancer data set, $F1$ would represent the average of our precision and recall.