MATH 390.4 / 650.2 Spring 2018 Homework #4t

Darshan Patel

Saturday 5th 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?

PECOTA is most similar to k- nearest neighbors.

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

Baseball performance as a function of age is not a linear model because peak performance doesn't change linearly with increasing age.

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

Baseball scouts can do better than a prediction system like PECOTA using a variety of skills. A pitcher can get many strikeouts by finding the strike zone and mixing up the pitches. Another way for analyzing player's potential is to have them do several tests on the field such as running and see how it plays out.

(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?

No one took advantage of Pitch f/x data to predict future success because no one knew how to make use of the data coming from Pitch f/x. People thought it may shift emphasis onto things that are harder to quantify and where the information is more exclusive.

(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.

The problem with weather predictions is that they are not accurate. Various factors, or xs, can change weather outcome for even in a small area. To predict weather, data has to be collected from multiple dimensions, or features, or zs, to create a full image. Another reason that weather is a dynamic system. It needs precise data to create predictions and sometimes data come truncated, since we cannot measure temperature and pressure to many decimal points.Even being off by 3 or 4 decimal places can throw off forecasts.

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

The weatherman lies about the chance of rain so that the viewer sees value in the perception of accuracy of weather predictions by for-profit weather forecasters. If a pro-profit forecaster said 50% chance of rain, it would seem wishy-washy and indecisive to people watching the forecast.

(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.

The problem with earthquake predictions is that it's done using information from past earthquakes which has a lot of noise, ϵ. From this, the models tend to overfit and thus create bad performance in the real world.

(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 nonsense predictor in the model Silver describes is color, which corresponds to a specific lock combination.

(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?

With four parameters, Neumann can create a model that will not overfit data points (of an elephant). With the addition of one more parameter, he can fine tune the parameters to give it a closer fit to the data. Although that would be nice, it cause overfitting and the elephant's trunk can lie anywhere in a small area.

(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.

The problem with unemployment predictions is that there are three challenges to overcome when making economic forecasts. It is hard to determine cause and effect from economic statistics alone. The economy is always changing so explanations of economic behavior that hold in one business cycle may not apply to future ones. Lastly, the data that economists work with isn't the best either.

(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.

Theory of how things function is important to have to make good predictions because it can allow for some accuracy. If you know what patterns can lead to rain, or a decrease in unemployment, then it will be easier for you to predict them.

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) Divide $\mathcal{D}$ into two parts: $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$.

(b) Create the model g from $\mathcal{H}$ on $\mathcal{D}_{\text{train}}$.

(c) Validate the model on $\mathcal{D}_{\text{test}}$ and calculate out of sample error.

(d) Ship the model g as well as the out of sample error.

(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) Divide $\mathcal{D}$ into $\mathcal{D}_{\text{test}}$ and $\mathcal{D}_{\text{train}}$ where $\frac{1}{k}$ of $\mathcal{D}$ is in $\mathcal{D}_{\text{test}}$.

(b) Fit $g_k = \mathcal{A}(\mathcal{H}, \mathcal{D}_{\text{train},k})$.

(c) Calculate $\hat{\vec{y}} = g_{VC}(x_{\text{test},k})$.

(d) Repeat 2-3 for $1 - K$ folds.

(e) Concat vertically $\hat{\vec{y}} = \begin{bmatrix} \hat{\vec{y}}_1 \\ \vdots \\ \hat{\vec{y}}_k \end{bmatrix}$.

(f) Compute $oose = \text{error}(\vec{y}, \hat{\vec{y}})$ where $\vec{y}$ is the full y since each observation is represented across the k folds.

(g) Ship g_k and the out of sample error $oose$.

(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) Divide $\mathcal{D}$ into $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{test}}$ and $\mathcal{D}_{\text{select}}$ where $\frac{1}{k}$ of $\mathcal{D}$ goes into $\mathcal{D}_{\text{test}}$ and $\mathcal{D}_{\text{select}}$.

(b) For $g_{j,k_i,k_0} = \mathcal{A}(\mathcal{H}, \mathcal{D}_{\text{train},k_i,k_0})$.

(c) Compute $\hat{y}_{j,k_i,k_0} = g_{j,k_i,k_0}(\mathcal{D}_{\text{select},k_i,k_o})$.

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

(e) Repeat steps 2-3 for all inner folds $k_i \in \{1, \ldots, 5\}$.

(f) Concat $\vec{\hat{y}}_{j,k_0} = \begin{bmatrix} \vec{\hat{y}}_{j,1,k_0} \\ \vdots \\ \vec{\hat{y}}_{j,5,k_o} \end{bmatrix}$.

(g) Select the best model $j^*_{k_0} = \arg\min \; oose_{j,k_0}$.

(h) Repeat steps 2-7 for $k_0 = \{1, \ldots, 5\}$.

(i) Get $\vec{\hat{y}} = \begin{bmatrix} \vec{y}_{j^*_1} \\ \vdots \\ \vec{y}_{j^*_5} \end{bmatrix}$.

(j) Estimate $oose = \text{error}(\vec{\hat{y}}, \vec{y})$.

(k) Repeat steps 2-7 to build final model g without $\mathcal{D}_{\text{test}}$.

(l) Ship g and out of sample error $oose$.

(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) Divide $\mathcal{D}$ into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ where $\frac{1}{k}$ of $\mathcal{D}$ is in $\mathcal{D}_{\text{test}}$.

(b) For each model $j \in \{1, \ldots, M\}$, build $g_j = \mathcal{A}_j(\mathcal{H}_j, \mathcal{D}_{\text{train}})$.

(c) Get error $oose_j = \text{error}(y_{\text{select}}, g_j(x_{\text{select}}))$ (usually s_e).

(d) Repeat steps 2-3 for M models

(e) Select $j^* = \arg\min \; \{oose_1, \ldots, oose_M\}$.

(f) Compute $oose_{j^*} = \text{error}(y_{\text{test}}, g_{j^*}(x_{\text{test}}))$. This is the estimate of future performance.

(g) Repeat steps 2-5 on $\mathcal{D}$ to produce g_{final}.

(h) Ship g_{final} and $oose_{j^*}$.

(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) Divide $\mathcal{D}$ into k folds, where $\mathcal{D}_{\text{test}}$ gets the first piece and $\mathcal{D}_{\text{train}}$ gets the rest.

(b) Fit $g_{k,m} = \mathcal{A}(\mathcal{H}, \mathcal{D}_{\text{train},k})$.

(c) Calculate $\vec{\hat{y}} = g_{cv}(x_{\text{test},k})$.

(d) Repeat 2-3 for $1 - K$ folds.

(e) Concat vertically $\vec{\hat{y}} = \begin{bmatrix} \vec{\hat{y}}_1 \\ \vdots \\ \vec{\hat{y}}_k \end{bmatrix}$.

(f) Repeat 2-4 for $1 - M$ models, where each model's $\vec{\hat{y}}$ go on its own column.

(g) Compute $oose = \text{error}(\vec{y}, \hat{\vec{y}})$ where $\vec{y}$ is the full y since each observation is represented across the k folds.

(h) Ship g_k and the out of sample error $oose$.

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 OLS solution would break down because there are more features being measured, to estimate the phenomenon, then the size of the dataset. When this happens, there are more unknown variables than equations and so there will be infinite solutions to the OLS algorithm.

(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".

$$\begin{aligned}
\lambda \left|\left|\boldsymbol{w}\right|\right|^2 &= \lambda \cdot \sum w_i^2 \\
&= \lambda \cdot \begin{bmatrix} w_1 & w_2 & \dots & w_n \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \\
&= \begin{bmatrix} \lambda w_1 & \lambda w_2 & \dots & \lambda w_n \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \\
&= \begin{bmatrix} w_1 & w_2 & \dots & w_n \end{bmatrix} \begin{bmatrix} \lambda w_1 \\ \lambda w_2 \\ \vdots \\ \lambda w_n \end{bmatrix} \\
&= \begin{bmatrix} w_1 & w_2 & \dots & w_n \end{bmatrix} (\lambda \boldsymbol{I}) \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \\
&= \boldsymbol{w}^\top(\lambda \boldsymbol{I})\boldsymbol{w}
\end{aligned}$$

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

$$\mathcal{H} = \left\{\boldsymbol{w} \cdot \boldsymbol{x} : \boldsymbol{w} \in \mathbb{R}^{p+1}\right\}$$

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

$$\begin{aligned} SSE &= (\boldsymbol{y} - \hat{\boldsymbol{y}})^\top (\boldsymbol{y} - \hat{\boldsymbol{y}}) \\ &= (\boldsymbol{y} - \boldsymbol{w} \cdot \boldsymbol{X})^\top (\boldsymbol{y} - \boldsymbol{w} \cdot \boldsymbol{X}) \\ &= \boldsymbol{y}^\top \boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{w}^\top \boldsymbol{y} - \boldsymbol{y}^\top \boldsymbol{X} \boldsymbol{w} + \boldsymbol{w}^\top \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{w} \end{aligned}$$

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

$$\begin{aligned} SSE &= \boldsymbol{y}^\top \boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{w}^\top \boldsymbol{y} - \boldsymbol{y}^\top \boldsymbol{X} \boldsymbol{w} + \boldsymbol{w}^\top \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{w} + \lambda \boldsymbol{w}^\top \boldsymbol{w} \\ &= \boldsymbol{y}^\top \boldsymbol{y} - 2\boldsymbol{w}^\top \boldsymbol{X}^\top \boldsymbol{y} + \boldsymbol{w}^\top (\boldsymbol{X}^\top \boldsymbol{x} + \lambda \boldsymbol{I}) \boldsymbol{w} \end{aligned}$$

(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.

When this is being minimized, the amount of deviation from the true phenomenon is minimized. The second part of this error function prevents overfitting by minimizing the size of $\boldsymbol{w}$.

(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.

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

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

$$\begin{aligned} SSE = -2\boldsymbol{X}^\top \boldsymbol{y} + 2(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}) \boldsymbol{w} &\overset{\text{set}}{=} 0 \\ 2(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}) \boldsymbol{w} = 2\boldsymbol{X}^\top \boldsymbol{y} \end{aligned}$$

The matrix $\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}$ needs to 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 $\forall \boldsymbol{z} \neq \boldsymbol{0} \;\; \boldsymbol{z}^\top A \boldsymbol{z} > 0$ then A is positive definite. Prove this matrix from the previous question is positive definite.

The quadratic form of the above matrix is

$$\boldsymbol{z}^\top (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}) \boldsymbol{z}$$

This simplifies to

$$\boldsymbol{z}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{z} + \boldsymbol{z}^\top \lambda \boldsymbol{I}\boldsymbol{z}$$

Note that $\boldsymbol{X}^\top \boldsymbol{X} = \sum x_i^2$, which is always positive, and so $\boldsymbol{z}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{z}$ is positive. Now we assumed that λ is a positive constant, therefore $\boldsymbol{z}^\top \lambda \boldsymbol{I}\boldsymbol{z}$ is also positive. Thus $\boldsymbol{z}^\top(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})\boldsymbol{z}$ is positive and so $\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}$ 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.

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

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

By adding the ridge penalty, we were able to figure out a way out of Luis's situation by not letting the model overfit through ridge regression.

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

Many of the values of the entries of $\boldsymbol{b}_{\text{ridge}}$ are close to 0 because the purpose of tacking on the ridge penalty was to limit the size of the $\boldsymbol{w}$. Ridge regression controlled the size of the constants.

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

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

$\hat{\boldsymbol{y}}$ is an orthogonal 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].

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

Attempt model selection protocol where each model runs on $\mathcal{D}_{\text{train}}$ and uses a different value of λ. Then examine which λ allows the smallest oose on $\mathcal{D}_{\text{test}}$. Cross validation is also another good method.

(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!

Attempt model selection protocol where each model runs on $\mathcal{D}_{\text{train}}$ and uses a different value of λ. Then examine which λ allows the smallest oose on $\mathcal{D}_{\text{test}}$. Cross validation is also another good method.

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

Ridge regression will also help when working with bad outliers in a dataset.

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

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

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 $\boldsymbol{b}_{\text{lasso}}$ 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 will be useful to get rid of predictors that cause prediction errors. If $\boldsymbol{b}_{\text{lasso},i}$ is 0 for a certain x, then the predictor will not be used to create the model. By doing so, more "important" predictors that play a bigger role in determining y will be looked at to create the model.

(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.

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?

These models came from parametric regressions of varying degrees and interaction terms.

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

The weakness in using M pre-specified models is that it is given a certain idea of how a phenomenon is determined. What this means is that a specific $\mathcal{H}$ is fed into the algorithm to produce M pre-specified models. It does not allow for interpretations to be made as the model is being made.

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

On $\mathcal{D}_{\text{train}}$, create a model using $\mathcal{H} = \{w_0 : w_0 \in \mathbb{R}\}$ and test it on $\mathcal{D}_{\text{select}}$. Then add a term that depends on a single x, forming $\mathcal{H} = \{w_0 + w_1 x : \boldsymbol{w} \in \mathbb{R}^2\}$. Create a model using this $\mathcal{H}$ and then test on $\mathcal{D}_{\text{select}}$. Continue adding terms until there is overfit in $\mathcal{D}_{\text{select}}$.

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

Start with a $\mathcal{H}$ of many linear terms. Create a model using $\mathcal{H}$ on $\mathcal{D}_{\text{test}}$ and test it on $\mathcal{D}_{\text{select}}$. Remove a term from $\mathcal{H}$ and create the model on $\mathcal{D}_{\text{train}}$ again. Test it on $\mathcal{D}_{\text{select}}$. Continue removing term until there is underfit in $\mathcal{D}_{\text{select}}$.

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

The weaknesses in the stepwise procedure is that you still need an intelligent collection of predictions to choose from at each position and you cannot be sure you are specifying it. Furthermore, the model will still be linear, perhaps nonlinear models can work better.

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

Non-parametric regression is regression using a $\mathcal{H}$ that can adjust flexibly and allow for complexity as the size of the data increases. Non-parametric regression models do not take a pre-specified form, thus solving the problem of nonadjustable $\mathcal{H}$. In addition, it constrains itself to information in the data, meaning it uses the data to make estimates. The goal of non-parametric regression is to estimate a phenomenon using only its data which provides the model space.

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

(a) Begin with all training data X, Y.

(b) For every possible split at the current node, divide into $X_L, \vec{y}_L$ and $X_R, \vec{y}_R$ and calculate

$$SSE_L = \sum (Y_L - \bar{Y}_L)^2 \quad SSE_R = \sum (Y_R - \bar{Y}_R)^2$$

where the summation is done over number of data points in the split.

(c) Find the split with the lowest total SSE

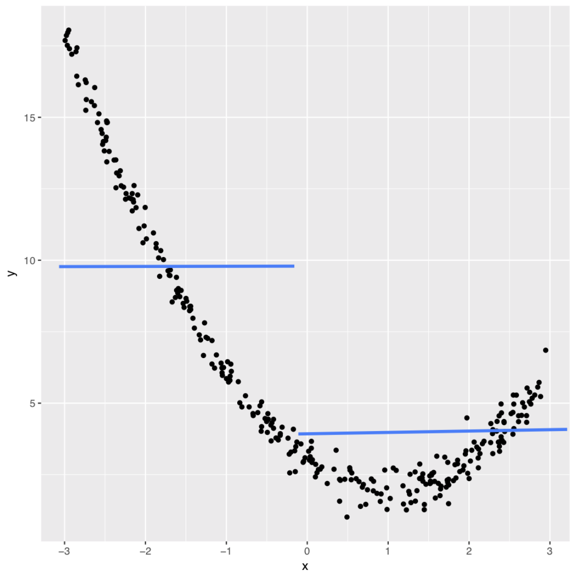
$$SSE_{tot} = SSE_L + SSE_R$$

(d) Create the split by splitting data into two nodes.

(e) Repeat steps 2-4 until "STOP."

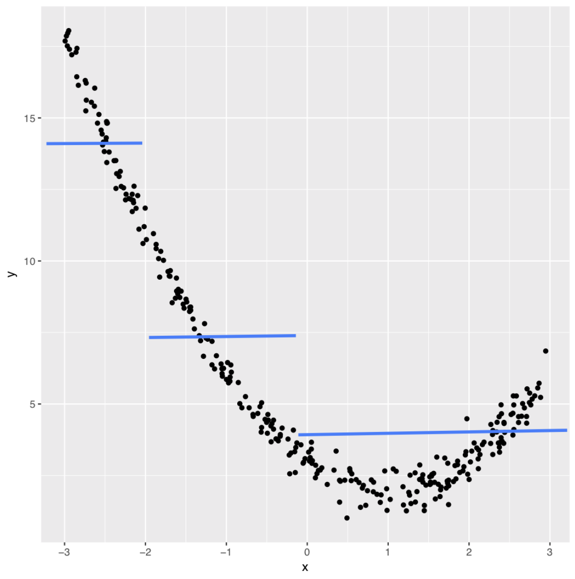
Note: STOP is when mode has less than N_0 data points inside.

(h) [easy] Consider the following data

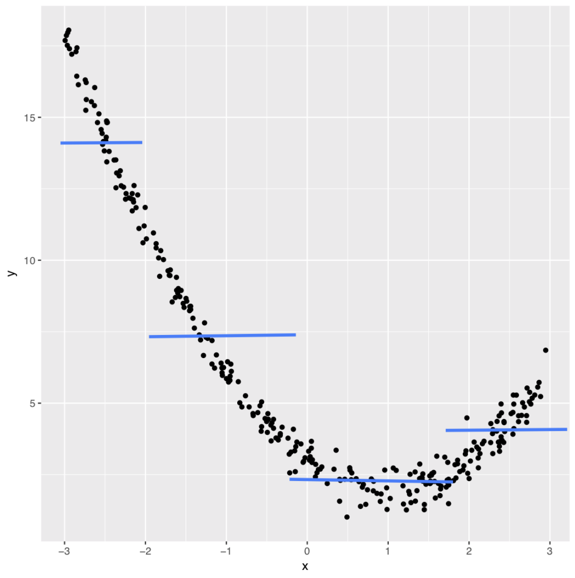


Create a tree with maximum depth 1 (i.e one split at the root node) and plot g above.

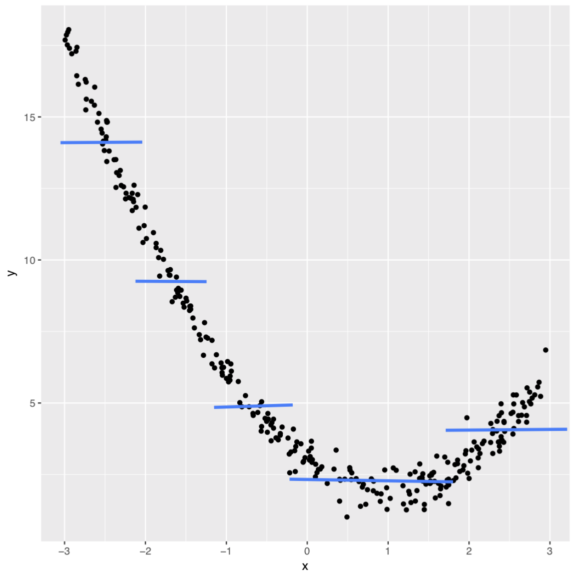
(i) [easy] Now add a second split to the tree and plot g below.



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



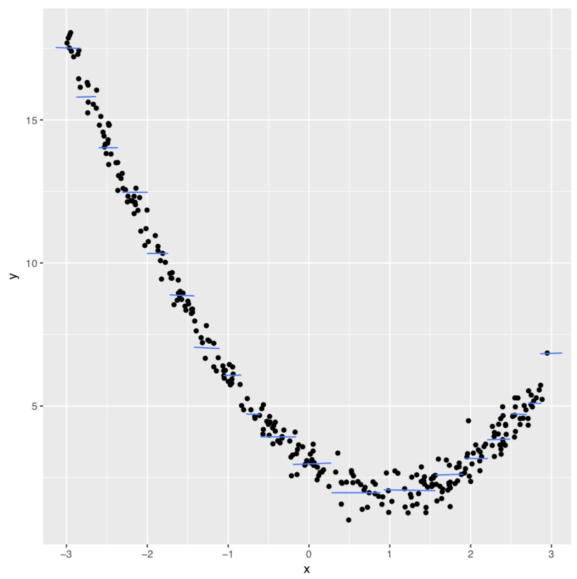
(k) [easy] Now add a fourth split to the tree and plot g below.



(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.

$$\begin{tikzcd}
 & & & & x < -0.20 \arrow[dll, no head] \arrow[drr, no head] & & & \\
 & & x < -1.15 \arrow[dl, no head] \arrow[dr, no head] & & & & x < 1.77 \arrow[dl, no head] \arrow[dr, no head] & \\
 & x < -2.10 \arrow[dl, no head] \arrow[dr, no head] & & \mathbf{4.9} & & \mathbf{2.4} & & \mathbf{4.0} \\
\mathbf{14.3} & & \mathbf{9.2} & & & & &
\end{tikzcd}$$

(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?

If $N_0 = 1$, there will be overfitting.

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

Create several models using a variety of N_0 and training on $\mathcal{D}_{\text{train}}$. Then test the model using $\mathcal{D}_{\text{test}}$ and calculate a out of sample error. Repeat this for all models and see which model's N_0 value created the smallest out of sample error.

Problem 5

These are questions about classification trees.

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

Classification trees and regression trees predict different types of phenomenon. Classification trees can predict a discrete $y \in \{1, \ldots, K\}$, where y can be categorical, whereas a regression tree is trying to find predict a $y \subseteq \mathbb{R}$.

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

(a) Begin with all training data.

(b) For every possible split, calculate the Gini impurity metrics

$$\text{Gini}_L = \sum_{i=1}^{k} \hat{p}_L(1 - \hat{p}_L)$$

$$\text{Gini}_R = \sum_{i=1}^{k} \hat{p}_R(1 - \hat{p}_R)$$

where

$$\hat{p}_i = \frac{\text{number of } y_i \text{ in category i}}{n \text{ number in node}}$$

(c) Find the split with the lowest weighted average of Gini impurity metric

$$\text{Gini}_{avg} = \frac{n_L \text{Gini}_L + n_R \text{Gini}_R}{n_L + n_R}$$

(d) Create the split and split the data in the node correctly along the two new daughter nodes.

(e) Repeat steps $2 - 4$ and “stop" where node has $\leq N_0$ data points. (Default is $N_0 = 1$.)

(f) For all leaf nodes, assign $\hat{y} = \text{Mode}[\vec{y}_0]$ where $\hat{y}_0$ 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 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. Each column represents the instances in a predicted value whereas each row represents the instances in the actual data.

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) $= 3475 + 262 + 471 + 792 = 5000$

FP (false positives) $= 262$

TP (true positives) $= 792$

FN (false negatives) $= 471$

TN (true negatives) $= 3475$

$\#P$ (number positive) $= 471 + 792 = 1263$

$\#N$ (number negative) $= 3475 + 262 = 3737$

$\#PP$ (number predicted positive) $= 262 + 792 = 1054$

$\#PN$ (number predicted negative) $= 3475 + 471 = 3946$

$\#P/n$ (prevalence / marginal rate / base rate) $= \frac{1263}{5000} = 0.2526$

$(FP + FN)/n$ (misclassification error) $= \frac{262+471}{5000} = 0.1466$

$(TP + TN)/n$ (accuracy) $= \frac{792+3475}{5000} = 0.8534$

$TP/\#PP$ (precision) $= \frac{792}{1054} = 0.7514$

$TP/\#P$ (recall, sensitivity, true positive rate, TPR) $= \frac{792}{1263} = 0.6270$

$2/(\text{recall}^{-1} + \text{precision}^{-1})$ (F1 score) $= \frac{2}{(792/1263)^{-1}+(792/1054)^{-1}} = \frac{1584}{2317} = 0.6836$

$FP/\#PP$ (false discovery rate, FDR) $= \frac{262}{1054} = 0.2485$

$FP/\#N$ (false positive rate, FPR) $= \frac{262}{3737} = 0.0701$

$FN/\#PN$ (false omission rate, FOR) $= \frac{471}{3946} = 0.1193$

$FN/\#P$ (false negative rate, FNR) $= \frac{471}{1263} = 0.3729$

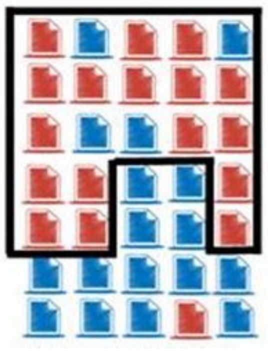
(c) [easy] Why is FPR also called the "false alarm rate"?

FPR is also called the false alarm rate because it tells how likely was a 1 detected even if there was a 0.

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

FNR is also called the miss rate because it tells how likely was a 0 detected when there was a 1.

(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.

The confusion table is as follows:

	0	1
0	13	4
1	1	17

Then

$$\text{precision} = \frac{TP}{PP} = \frac{17}{21} = 0.8095$$

and

$$\text{recall} = \frac{TP}{P} = \frac{17}{18} = 0.9444$$

(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.

A situation where it'll be profitable to look at FPR vs. FNR is cancer detection. It will be better to have a good high FPR value and low FNR value. We want people to be wary in the off chance they do in fact have cancer.

(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.

A situation where it'll be profitable to look at FDR vs. FOR is cancer detection. It will be better to have a high FDR value and low FOR value. We want people to be wary that have cancer rather than never finding out.

(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.

A situation where it'll be profitable to look at precision vs. recall is when doing pattern recognition, such as identifying how people will vote in an upcoming election. We want to know whether people will stick to their political party candidate or switch to the opposing party.

(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.

A situation where it'll be profitable to look at an overall metric such as accuracy is when evaluating the efficiency of an cancer detector. It will be good to have both precision and recall high as possible so that cancer can be properly detected.