MATH 342W / 650.4 Spring 2022 Homework #2

Peter Antonaros

Due 11:59PM Thurs, Mar 3 by email

(this document last updated 4:40am on Friday 4th March, 2022)

Instructions and Philosophy

The path to success in this class is to do many problems. Unlike other courses, exclusively doing reading(s) will not help. Coming to lecture is akin to watching workout videos; thinking about and solving problems on your own is the actual "working out." Feel free to "work out" with others; **I want you to work on this in groups.**

Reading is still *required.* You should be googling and reading about all the concepts introduced in class online. This is your responsibility to supplement in-class with your own readings.

The problems below are color coded: green problems are considered *easy* and marked "[easy]"; yellow problems are considered *intermediate* and marked "[harder]", red problems are considered *difficult* and marked "[difficult]" and purple problems are extra credit. The *easy* problems are intended to be "giveaways" if you went to class. Do as much as you can of the others; I expect you to at least attempt the *difficult* problems.

This homework is worth 100 points but the point distribution will not be determined until after the due date. See syllabus for the policy on late homework.

Up to 7 points are given as a bonus if the homework is typed using LaTeX. Links to instaling LaTeX and program for compiling LaTeX is found on the syllabus. You are encouraged to use `overleaf.com`. If you are handing in homework this way, read the comments in the code; there are two lines to comment out and you should replace my name with yours and write your section. The easiest way to use overleaf is to copy the raw text from hwxx.tex and preamble.tex into two new overleaf tex files with the same name. If you are asked to make drawings, you can take a picture of your handwritten drawing and insert them as figures or leave space using the "\vspace" command and draw them in after printing or attach them stapled.

The document is available with spaces for you to write your answers. If not using LaTeX, print this document and write in your answers. I do not accept homeworks which are *not* on this printout. Keep this first page printed for your records.

NAME: ______________________Peter Antonaros______________________

Problem 1

These are questions about Silver's book, chapter 2, 3. Answer the questions using notation from class (i.e. $t, f, g, h^*, \delta, \epsilon, e, t, z_1, \ldots, z_t, \mathbb{D}, \mathcal{H}, \mathcal{A}, \mathcal{X}, \mathcal{Y}, X, y, n, p, x_{\cdot 1}, \ldots, x_{\cdot p}, x_{1\cdot}, \ldots, x_{n\cdot}$, etc).

(a) [harder] If one's goal is to fit a model for a phenomenon y, what is the difference between the approaches of the hedgehog and the fox? Connecting this to the modeling framework should really make you think about what Tetlock's observation means for political and historical phenomena.

The main difference between a "hedgehog" and a "fox" is how they see the world.

Hedgehog: Believes the world (phenomena) can be simplified to a few governing laws.

Fox: Believes the world (phenomena) are inherently complex, and nearly impossible to simulate.

This means that generally foxes produce more accurate models, because they attempt to account for complexity, whereas a hedgehog tries to identify a simple signal through a lot of noise. Foxes think probabilistically and accept the fact that there will be error regardless of features accounted for. Hedgehogs seek a clear and orderly solution which often times is incorrect and leads to false confidence in their predictions. In the context of political and historical phenomena Tetlock's observations mean that is is nearly impossible to model and pinpoint reasoning behind action in this realm. There are simply too many factors to account for, many of which are totally impossible to know with certainty. This means when we speak about historical events and or politics the best we can do is speak the probable language. We don't know why person X won the election, and we don't know why person Y invaded country Z.

(b) [easy] Why did Harry Truman like hedgehogs? Are there a lot of people that think this way?

Harry Truman liked hedgehogs because of the environment he was leading in. There was little room for "maybes" and "probabilities", he needed certainty. Leading during the runup to the Cold War, the last thing a president wants to hear is "there is a 30 percent chance the USSR will use a nuclear weapon, a 30 percent chance they won't, and a 40 percent chance of war". Politicians need certainty that way they can clearly convey a message to the people they are leading. Unfortunately this leads to wrong answers and wrong predictions, but sometimes false certainty is better than true probability when attempting to convince the public they are safe.

There are many people that think like hedgehogs and I would argue it is the norm for humans to think in this way. Back when humans were getting mauled by packs of wild dogs, there was no time in thinking of how probable it was to get eaten. Either we were going to be hunted or we were going to be the hunter. Now that we have transitioned and essentially dominated the environment we can begin to think more like a fox because safety has allowed us this luxury. Still though it is ingrained in us to want specific answers, rather than "up in the air" probabilities.

(c) [difficult] Why is it that the more education one acquires, the less accurate one's predictions become?

This seems somewhat counter intuitive, but as you learn more, you begin to realize the sheer amount of things that are unknown to you. So what does this mean for predictions? It means that the more you educate yourself about the world, when you attempt to model it; A) You are accounting for greater levels of complexity, B) You try to model phenomena that are inherently more complex.

Thousands of years ago, people attempted to model the movement of the sun and use this as a basis for predicting (knowing) the time. Now, we would never accept this as within our error bounds simply because the equipment was imprecise. Now we use radioactive decay as a form of time elapsed measurements and as we learn more about quantum mechanics we realize even this is still not truly accurate, and we have not accounted for that.

To sum up this answer I'd like to say as mankind progresses in modeling, we use a larger number of features and more accurate ones, but we realize there may be a "built in" randomness that we cannot overcome.

(d) [easy] Why are probabilistic classifiers (i.e. algorithms that output functions that return probabilities) better than vanilla classifiers (i.e. algorithms that only return the class label)? We will move in this direction in class soon.

They allow for a larger output space, which ultimately models the real world in a more accurate way. Nature is not black and white, and attempting to simply categorize things based on vanilla classifiers is rudimentary.

Probabilistic classifiers also allow for new information to be incorporated into the model quite easily. Imagine we are looking at a picture of a bug and our model is saying it is a "German Roach". There could be a time in the future where science discovers it's some new species of Roach. Our probabilistic classifier would have accounted for this by saying "I am X percent certain is a German Roach", leaving 1-X percent to the discovery or input of more information.

P classifiers account for uncertainty and randomness, where V classifiers do not.

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

PECOTA is most similar to k Nearest Neighbors. It measures how "close" is a prospect player's stats "features" are to that of major league players and classifies it as one of 13 possible career paths. I can see we have some sort of distance function that then classifies a player into some number of groupings, and so k Nearest Neighbors would be my guess for how PECOTA works.

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

No baseball performance is not a linear function of age. The reason being, a player generally starts off with base skill, gradually increases there skill until they become older where it begins to drop off again. This is also backed by the image on page 83 of the book. The data is noisy but we can clearly see the function is not linear, but much rather the approximating function is in the $\mathcal{H}$ space of Quadratic functions.

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

Baseball scouts can do better than a prediction system like PECOTA by evaluating hard to quantify features about a player. An example would be, PECOTA didn't take into account Pedroia's attitude towards the game and the people around him. Despite his size/strength and so on, none of which were really in his favor he was not bothered by negative coverage, negative comments, and genuinely wanted to do his best. For scouts this would be something that is quite easy to see, but for a system like PECOTA, quantifying attitude can be close to impossible. Scouts can look beyond just the metrics and stats of a prospect, seeing them for who they are; a human with emotion, with ups and downs, with slumps and streaks. This isn't to say that PECOTA is not an effective system, but there are cases in which the metrics are stacked against a player and yet they still play the game well. Scouts are perfect for this situation, because they as humans can use their innate ability to judge the "feel" of a situation. This "feel" is something we cannot even define ourselves, we just know it when we see it, and so it would be a monumental challenge to implement in a model like PECOTA.

(h) [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?

At the time of the writing of Silver's book no one has taken advantage of the Pitch f/x data because it is difficult to fuse quantitative and qualitative evaluations about a player together. Pitch f/x has a ton of data, much of which is probably noise and needs to be cut down to size, into meaningful chunks. In addition, there needs to be ways to incorporate this with qualitative evaluations such as stress management, time management, attitude towards critique and so on. At the end of the day we are trying to model humans, and we cannot do so simply with data. If humans could be perfectly modeled with quantitative data then we would be robots. This is why no one at the time has attempted to model with Pitch f/x because no one has found a meaningful way to represent and quantify metrics that are inherently qualitative and often times subjective.

How do we measure a player's attitude? We can't say they have a 2.3 attitude score like we can a 2.3 ERA and for this exact reason no one (at the time) has taken advantage of Pitch f/x data. The data alone is not the full story and won't produce models with the accuracy demanded by the sport.

Problem 2

These are questions about the SVM.

(a) [easy] State the hypothesis set $\mathcal{H}$ inputted into the support vector machine algorithm. Is it different than the $\mathcal{H}$ used for $\mathcal{A}$ = perceptron learning algorithm?

Hypothesis Set for Support Vector Machine

$$\mathcal{H} = \{ \mathbb{1}_{\vec{w}.\vec{x}-b \geq 0} : \vec{w} \in \mathbb{R}^p, b \in \mathbb{R}$$

$\vec{w} \cdot \vec{x} - b = \emptyset \Rightarrow$ Hesse Normal Form of a line

Hypothesis Set for Perceptron Learning

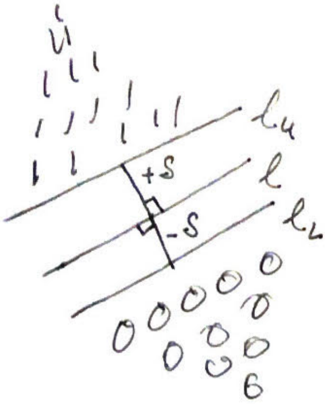
$\mathcal{H} = \{ \mathbb{1}\ \vec{w} \cdot \vec{x} \geq 0 ; \vec{w} \in \mathbb{R}^p \} \Rightarrow$ We "over parameterized" the model/

$$= \mathbb{1}_c (\vec{w} \cdot \vec{x}) \geq 0 . \vec{w} \in \mathbb{R} , \forall c \in \mathbb{R}$$

Support Vector Machine includes a real value b

(b) [E.C.] Prove the max-margin linearly separable SVM converges. State all assumptions. Write it on a separate page.

(c) [difficult] Let $\mathcal{Y} = \{-1, 1\}$. Rederive the cost function whose minimization yields the SVM line in the linearly separable case.



$$\forall_i y_i = 1;\ \vec{\omega}\cdot\vec{x}_i - (b+1) \geq 0$$

$$\Rightarrow \vec{\omega}\cdot\vec{x}_i - b \geq 1$$

$$\Rightarrow \underline{y_i(\vec{\omega}\cdot\vec{x}_i - b) \geq 1}$$

$$\forall_i y_i = -1;\ \vec{\omega}\cdot\vec{x}_i - (b-1) \leq 0$$

$$\Rightarrow \vec{\omega}\cdot\vec{x}_i - b \leq -1$$

$$\Rightarrow \underline{y_i(\vec{\omega}\cdot\vec{x}_i - b) \geq 1}$$

(d) [easy] Given your answer to (c) rederive the cost function using the "soft margin" i.e. the hinge loss plus the term with the hyperparameter λ. This is marked easy since there is just one change from the expression given in class.

$$SHE = \sum_{i=1}^{n} H_i \quad ; \quad H_i = \max\{0, 1 - y_i(\vec{\omega}\cdot\vec{x}_i - b)\}$$

Minimizing Soft Margin $\equiv$ Minimizing SHE

$$\operatorname{argmin}_{\vec{\omega}, b}\left\{\frac{1}{n} SHE + \lambda\|\vec{\omega}\|^2\right\} \Rightarrow g = \mathcal{A}(\mathbb{D}, \mathcal{H})$$

Our algorithm will minimize this!

Problem 3

These are questions are about the k nearest neighbors (KNN) algorithm.

(a) [easy] Describe how the algorithm works. Is k a "hyperparameter"?

Let's start off by saying that k is most certainly a hyper parameter. K has control over the learning process, and needs to be "tuned". The k value is dependent on factors of the individual data set but at the same time is independent of the model. By independent of the model I mean that k influences the model outcomes, rather than the other way around.

Briefly k Nearest Neighbors classifies data based on the distance between the data point in question and its K neighbors who minimize this distance.

1) Start off with some k value -> this will be tuned based on our results from training

2) For each row of $x_{\cdot 1}, \ldots, x_{\cdot p}$ in $\mathbb{D}$ calculate the distance between our query and the current row

3) Store and sort the distances (lowest to highest) in some ordered set

4) Pick the k smallest distances and return the mode of the k labels as the result for our query

(b) [difficult] [MA] Assuming $\mathcal{A}$ = KNN, describe the input $\mathcal{H}$ as best as you can.

I would say the the input space, $\mathcal{H}$ for KNN, is something similar to a density region. Ultimately we are looking for the "optimized density" region in order to minimize classification error.

k is the number of elements we are comparing against, let's say V is the volume of the region (assuming more than 2 features or else we could say A for area) and N is the total number of observations. Then we can say that the $\mathcal{H}$ is the space of all densities $[(k)/(V * N)]$

(c) [easy] When predicting on $\mathbb{D}$ with $k = 1$, why should there be zero error? Is this a good estimate of future error when new data comes in? (Error in the future is called *generalization error* and we will be discussing this later in the semester).

For $k = 1$ there will be zero error on the training data, large emphasis on training data. This is because if we assume our classifications to be correct, then the closest point to a point of some classification will have the same classification.

This is definitely not a good estimate on future error when new data comes in but it does provide a good baseline. It seems like to me this would be a good thing to say "if my KNN model can't beat a $k = 1$ model" then there is close to no point in even using this algorithm.

Problem 4

These are questions about the linear model with $p = 1$.

(a) [easy] What does $\mathbb{D}$ look like in the linear model with $p = 1$? What is $\mathcal{X}$? What is $\mathcal{Y}$?

Linear Model with $p=1$

$$\left.\begin{array}{l} \mathcal{X} = \text{Input Space} \\ \mathcal{Y} = \text{Output Space} \end{array}\right\} \Rightarrow \mathbb{D} = \text{Dataset}$$

→ One Feature $(n \times p)$

$$\overline{X} = \begin{bmatrix} \vec{x}_{11} \\ \vec{x}_{21} \\ \vdots \\ \vec{x}_{n1} \end{bmatrix} \qquad \overline{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad \Rightarrow \mathbb{D} = \left[\begin{array}{c|c} x_{11} & y_1 \\ x_{21} & y_2 \\ \vdots & \vdots \\ x_{n1} & y_n \end{array}\right]$$

Rowise Vectors $\odot$ Columnwise Vectors $\Rightarrow \mathbb{D} = \langle \vec{x}_i, \vec{y}_i \rangle = [\overline{X} \mid \overline{Y}]$

(b) [easy] Consider the line fit using the ordinary least squares (OLS) algorithm. Prove that the point $< \bar{x}, \bar{y} >$ is on this line. Use the formulas we derived in class.

Intuitively we can say that since the OLS algorithm produces a **continuous** line through the domain and range of our data, $< \bar{x}, \bar{y} >$ which represent the averages of x_i and y_i (respectively) will certainly be on the line itself.

Let $g(x)$ be the line produced by OLS and let $r = \frac{s_{xy}}{s_x s_y}$

Then we can say, with p=1, $g(x) = \bar{y} - r\frac{s_{xy}}{s_x s_y}\bar{x} + r\frac{s_{xy}}{s_x s_y}x = \bar{y} - r\frac{s_{xy}}{s_x s_y}(\bar{x} - x)$

If we let $x = \bar{x}$ and substitute into $g(x)$ above we can see that the resulting output is $\bar{y}$, and thus the point $< \bar{x}, \bar{y} >$ is on our OLS line.

(c) [harder] Consider the line fit using OLS. Prove that the average prediction $\hat{y}_i := g(x_i)$ for $x_i \in \mathbb{D}$ is $\bar{y}$.

(d) [harder] Consider the line fit using OLS. Prove that the average residual e_i is 0 over $\mathbb{D}$.

$e_i = y_i - \hat{y}_i = \frac{1}{n} * \sum_{i=1}^{n}(y_i) - \frac{1}{n} * \sum_{i=1}^{n}(\hat{y}_i) = \bar{y} - \bar{y} = 0$

(e) [harder] Why is the RMSE usually a better indicator of predictive performance than R^2? Discuss in English.

Generally RMSE is a better indicator of predictive performance than R^2 for the following reasons...

1) Easily interpretable (units are the same as the response variable)

2) Error independent of the null model. R^2 measures error in relation to the null model ($g0$). This is sometimes helpful but usually we want to see how our error performance irrespective of $g0$

(f) [harder] R^2 is commonly interpreted as "proportion of the variance explained by the model" and proportions are constrained to the interval $[0, 1]$. While it is true that $R^2 \leq 1$ for all models, it is not true that $R^2 \geq 0$ for all models. Construct an explicit example $\mathbb{D}$ and create a linear model $g(x) = w_0 + w_1 x$ whose $R^2 < 0$.

$R^2 = 1 - (SSE/SST)$, from this we can see that it is not necessarily true that $R^2 \geq 0$ because we can create a model so awful that it's error is greater than $g0$ and thus $SSE \geq SST$. This means we have $1 - num \leq 0 : num \geq 1$.

Example:

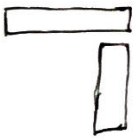
Let $g0$ be the null model such that its predicted values are $\bar{y}$ (mean) and $g(x) = 0 + 1x$

Let $\mathbb{D} =< \vec{X}, \vec{Y} >$, such that $\vec{X} =< 1, 2, 3, 4, 5 >$ and $\vec{Y} =< 1, 2, 3, 4, 5 >$, and so our null model will predict 2.5

Our $R^2 = \frac{\sum_{i=1}^{5}(y_i - (0 + x_i))^2}{\sum_{i=1}^{5}(y_i - 2.5)^2} \leq 0$

(g) [difficult] You are given $\mathbb{D}$ with n training points $< x_i, y_i >$ but now you are also given a set of weights $[w_1 \; w_2 \; \ldots \; w_n]$ which indicate how costly the error is for each of the i points. Rederive the least squares estimates b_0 and b_1 under this situation. Note that these estimates are called the *weighted least squares regression* estimates. This variant $\mathcal{A}$ on OLS has a number of practical uses, especially in Economics. No need to simplify your answers like I did in class (i.e. you can leave in ugly sums).

$\vec{W} = [w_1, w_2, w_3, \ldots, w_n]$ (Row Vector)



$\vec{E} = [e_1, e_2, e_3, \ldots, e_n]$ (Column Vector)

$\vec{w} \cdot \vec{E}$ is our minimization target to find b_0 and b_1

$$\vec{w} \cdot \vec{E} = \sum_{i=1}^{n} w_i \cdot (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} w_i \cdot (y_i - b - cx_i)^2$$

$$= w_i(y_i^2 - 2y_i b - 2y_i + b^2 + 2bcx_i + c^2 x_i^2)$$

$$= \sum_{i=1}^{n} w_i y_i^2 - 2b\sum_{i=1}^{n} w_i y_i - 2\sum_{i=1}^{n} w_i y_i + b^2 \sum_{i=1}^{n} w_i + 2bc\sum_{i=1}^{n} w_i x_i + c^2 \sum_{i=1}^{n} w_i x_i^2$$

$$\frac{\partial[\vec{W} \cdot \vec{E}]}{\partial b} = 0 \quad \Big| \quad b_0 = \left(b_1 \Sigma_{i=1}^{n} w_i x_i - \Sigma_{i=1}^{n} w_i y_i\right) / \left(\Sigma_{i=1}^{n} w_i\right)$$

$$\frac{\partial[\vec{W} \cdot \vec{E}]}{\partial c} = 0 \quad \Big| \quad b_1 = \frac{(-\Sigma_{i=1}^{n} w_i y_i)(\Sigma_{i=1}^{n} w_i x_i)}{(\Sigma_{i=1}^{n} w_i)(\Sigma_{i=1}^{n} w_i x_i^2) - (\Sigma_{i=1}^{n} w_i x_i)^2}$$

use to solve b_0

(h) [harder] Interpret the ugly sums in the b_0 and b_1 you derived above and compare them to the b_0 and b_1 estimates in OLS. Does it make sense each term should be altered in this matter given your goal in the weighted least squares?

Yes because each beta is being altered by the respective w_i weights

(i) [E.C.] In class we talked about $x_{raw} \in \{\text{red}, \text{green}\}$ and the OLS model was the sample average of the inputted x. Imagine if you have the additional constraint that x_{raw} is ordinal e.g. $x_{raw} \in \{\text{low}, \text{high}\}$ and you were forced to have a model where $g(\text{low}) \leq g(\text{high})$. Write about an algorithm $\mathcal{A}$ that can solve this problem.

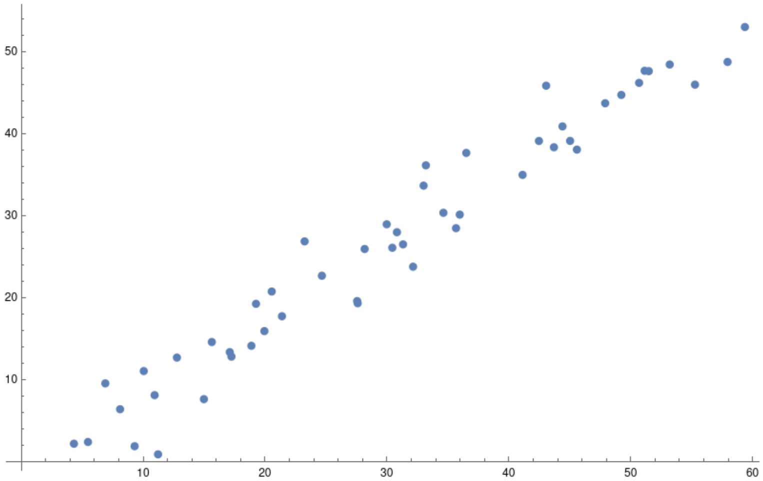
Problem 5

These are questions about association and correlation.

(a) [easy] Give an example of two variables that are both correlated and associated by drawing a plot.

Plot was made using Mathematica

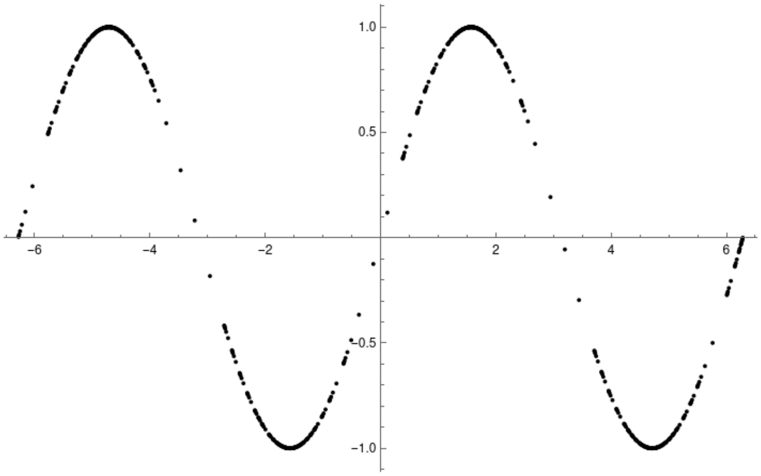
Here the x and y variables are associated by a linear line, and there is a positive correlation between the two.



(b) [easy] Give an example of two variables that are not correlated but are associated by drawing a plot.

Plot was made using Mathematica

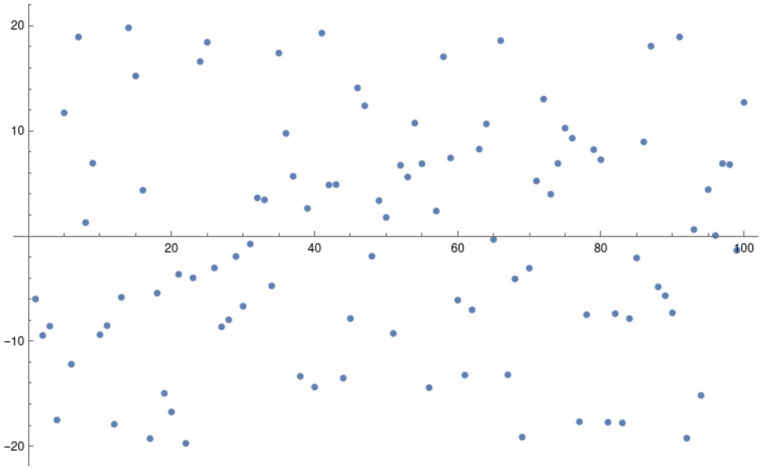
Here the x and y variables are associated by a Sine curve, but there is no correlation in the graph.



(c) [easy] Give an example of two variables that are not correlated nor associated by drawing a plot.

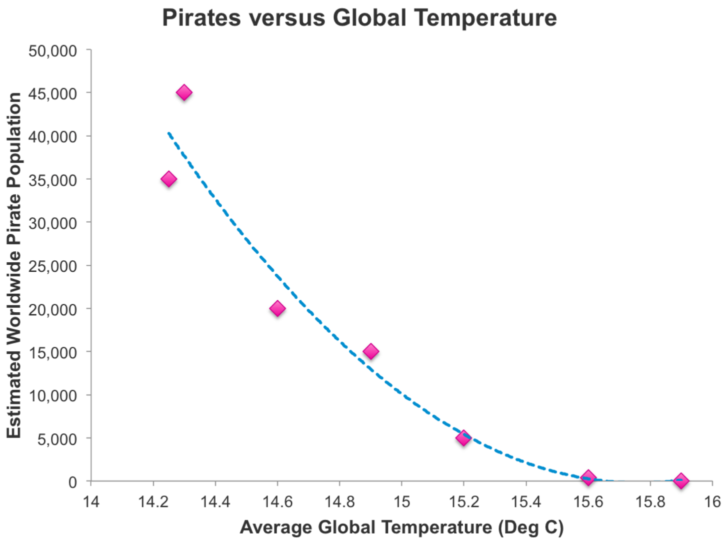
Plot was made using Mathematica

Here the x and y variables are neither associated nor correlated. They are simply randomly selected points.



(d) [easy] Can two variables be correlated but not associated? Explain.

Yes two variables can be correlated by not associated. Association implies dependence on one variable from another. This need not be true for correlation. Here is a funny example that shows a negative correlation with absolutely zero association between the two variables.



Problem 6

These are questions about multivariate linear model fitting using the least squares algorithm.

(a) [difficult] Derive $\frac{\partial}{\partial \boldsymbol{c}}\left[\boldsymbol{c}^\top A \boldsymbol{c}\right]$ where $\boldsymbol{c} \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n\times n}$ but *not* symmetric. Get as far as you can.

$\frac{\partial}{\partial c}[c^T A c]$: $c \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n\times n}$ (Not Symmetric)

$$\frac{\partial}{\partial c_1}[c^T A c] = \underline{2a_{11}c_1} + a_{12}c_2 + \dots + a_1 n c_n$$

$$\begin{array}{c} A \\ \boxed{\begin{array}{l} 11\ 12 \dots \\ 21\ 22 \dots \\ 31\ 32\ 33 \dots \\ n1 \dots nn \end{array}} \\ \text{NoSym} \end{array} \quad \begin{array}{c} C \\ \boxed{\begin{array}{c} c_1 \\ c_2 \\ | \\ c_n \end{array}} \\ \text{Col} \end{array}$$

$$\frac{\partial}{\partial c_2}[c^T A c] = a_{21}c_1 + \underline{2a_{22}c_2} + \dots + a_2 n c_n$$

$$\frac{\partial}{\partial c_3}[c^T A c] = a_{31}c_1 + a_{32}c_2 + \underline{2a_{33}c_3} + \dots + a_{3n}c_n$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$\frac{\partial}{\partial c_n}[c^T A c] = a_{n1}c_1 + a_{n2}c_2 + \dots + \underline{2a_{nn}c_n}$$

$$= 2 \cdot \sum A_{k,j} c_k \; ; k = j \text{ (on Diaganol)}$$

$$+$$

$$\sum A_{k,j} c_k \; ; k \neq j \text{ (Everywhere Else)}$$

(b) [easy] Given matrix $X \in \mathbb{R}^{n\times(p+1)}$, full rank and first column consisting of the $\mathbf{1}_n$ vector, rederive the least squares solution $\boldsymbol{b}$ (the vector of coefficients in the linear model shipped in the prediction function g). No need to rederive the facts about vector derivatives.

Matrix $X \in \mathbb{R}^{n\times(p+1)}$ => Full Rank with $\vec{1}_n$ vector in col 1.

$$X = \underbrace{\left[\vec{1}_n \;\vdots\; \vec{x}_1 \;\vdots\; \cdots \;\vdots\; \vec{x}_p\right]}_{p+1} \Big\updownarrow n \qquad \underline{\underline{\mathcal{A}_{OLS} \Rightarrow \frac{\partial}{\partial \vec{w}}[SSE] \overset{set}{:=} \vec{0}_{p+1}}}$$

$$\Rightarrow \frac{\partial}{\partial \vec{w}}\left[\vec{e}^{\,T} e\right] = \frac{\partial}{\partial \vec{w}}\left[(\vec{y} - \vec{\hat{y}})^T(\vec{y} - \vec{\hat{y}})\right]$$

$$= \frac{\partial}{\partial \vec{w}}\left[(\vec{y} - X\vec{w})^T(\vec{y} - X\vec{w})\right]$$

$$= \frac{\partial}{\partial \vec{w}}\left[(\vec{y}^{\,T} - \vec{w}^{\,T}X^T)(\vec{y} - X\vec{w})\right]$$

(Many simplification steps later)

$$\Rightarrow I_{p+1}\,\vec{w} = (X^TX)^{-1}X^T\vec{y} \Rightarrow \boxed{\vec{b}_{OLS} = (X^TX)^{-1}(X^T\vec{y})}$$

(c) [harder] Consider the case where $p = 1$. Show that the solution for $\boldsymbol{b}$ you just derived in (b) is the same solution that we proved for simple regression. That is, the first element of $\boldsymbol{b}$ is the same as $b_0 = \bar{y} - r\frac{s_y}{s_x}\bar{x}$ and the second element of $\boldsymbol{b}$ is $b_1 = r\frac{s_y}{s_x}$.

$$\vec{b}_{OLS} = (X^TX)^{-1}X^T\vec{y}$$

$$\vec{b} = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}, \quad b_0 = \bar{y} - r \cdot \frac{s_y}{s_x}\bar{x}, \quad b_1 = r \cdot \frac{s_y}{s_x}$$

$(X^TX)^{-1}X^T\vec{y}$ * Multiplying Through

$$\underbrace{\left(\begin{bmatrix} 1, 1, \ldots, 1_n \\ x_1, x_2, \ldots, x_n \end{bmatrix} \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1_n & x_n \end{bmatrix}\right)}_{A} \underbrace{\begin{bmatrix} 1, 1, \ldots, 1_n \\ x_1, x_2, \ldots, x_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{\vec{\gamma}}$$

$$\begin{bmatrix} n & \Sigma x_i \\ \Sigma x_i & \Sigma x_i^2 \end{bmatrix} \cdot \begin{bmatrix} \Sigma y_i \\ \Sigma x_i y_i \end{bmatrix} \equiv \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$$

(d) [easy] If X is rank deficient, how can you solve for $\boldsymbol{b}$? Explain in English.

Either you have linearly dependent columns and thus you'll need to get rid of those, or you are trying to find an estimate for n parameters with less than n data points and so you simply need more data.

(e) [difficult] Prove $\text{rank}\,[X] = \text{rank}\,\left[X^\top X\right]$.

Prove $\text{rank}[X] = \text{rank}[X^T X]$

Let $X \in \mathbb{R}^{n \times p}$, with $\text{rank}[X] = k$

$\hookrightarrow$ Let $X\vec{v} = \vec{0}$

$\Rightarrow \vec{v}^T X^T X \vec{v} = \vec{0}_n \Rightarrow \text{rank}[X^T X \vec{v}] = k$

$\Rightarrow X^T X \vec{v} = \vec{0}_n \rightarrow X^T X \vec{v} = X^T \vec{0}_n = \vec{0}_p$

$\Rightarrow X^T X \vec{v} = \vec{0}_p \rightarrow \vec{v}^T X^T \vec{v} = 0$

$\Rightarrow \vec{v}^T X^T \vec{v} = \langle X\vec{v} \rangle \langle X\vec{v} \rangle = 0 \rightarrow X\vec{v} = \vec{0}_n$

Thus, $\text{rank}[X] = \text{rank}[X^T X]$

(f) [harder] [MA] If $p = 1$, prove $r^2 = R^2$ i.e. the linear correlation is the same as proportion of sample variance explained in a least squares linear model.

(g) [harder] Prove that $g([1\ x_1\ x_2\ \ldots\ x_p]) = \bar{y}$ in OLS.

Prove $g[1, x_{\cdot 1}, x_{\cdot 2}, x_{\cdot 3}, \ldots, x_{\cdot p}] = \bar{y}$ in OLS

we know $g(\vec{x}^*) = \hat{y}^* = b_0 + b_1 x_1^* + \ldots + b_p x^*_p$

$\hookrightarrow g[1, x_{\cdot 1}, \ldots, x_{\cdot p}] = b_0 + b_1 \bar{x}_1 + \ldots + b_p \bar{x}_p$

$$= \frac{b_0}{n} \sum_i (1) + \frac{b_1}{n} \sum_i \bar{x}_1 + \ldots + \frac{b_p}{n} \sum_i \bar{x}_p$$

$$= \frac{1}{n} \left(b_0 * \sum_i (1) + b_1 \sum_i \bar{x}_1 + \ldots + b_p \sum_i \bar{x}_p \right)$$

$$= \frac{1}{n} (\hat{y}) = \frac{\hat{y}}{n} = \underbrace{\bar{\hat{y}}}_{\text{shown in Earlier Proof.}} \equiv \bar{y}$$

(h) [harder] Prove that $\bar{e} = 0$ in OLS.

We know that OLS minimizes SSE, i.e the squared distance between data point and OLS line is minimized by setting the derivative of SSE to 0. Intuitively we can say that this minimization results in a "mean line" through the data points. It then follows that given a mean line, through some data points, $\bar{e}$ will equal 0.

Prove $\vec{e} = 0$ in OLS

Let X be $n \times k$ matrix of regressors

Let $\vec{e}$ be the residual's vector

Let $\vec{y}$ be output vector

If we let $M = I_n - \overset{\text{Hat Matrix}\left(X(X^T X)^{-1} X^T\right)}{H}$

$M\vec{y} = \vec{e}$ 1) M is symmetric
2) MX is 0

Appending our $\vec{1}$ to X, we can say $i = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1_n \end{bmatrix}$

$$\sum_{i=1}^{n} \vec{e}_i = i^T \vec{e} = i^T M \vec{y}$$

$$= i^T M^T \vec{y} = (Mi)^T \vec{y} = 0^T \vec{y} = 0$$

we showed the sum of the residuals is 0 and so $\bar{e}$ being the average residual must be 0 if the sum e_i is 0.

(i) [difficult] If you model $\boldsymbol{y}$ with one categorical nominal variable that has levels A, B, C, prove that the OLS estimates look like $\bar{y}_A$ if $x = A$, $\bar{y}_B$ if $x = B$ and $\bar{y}_C$ if $x = C$. You can choose to use an intercept or not. Likely without is easier.

$$\vec{b} = \begin{bmatrix} \bar{y}_A \\ \bar{y}_B \\ \bar{y}_C \end{bmatrix} \Rightarrow \text{Show This}$$

$$\vec{b} = (X^T X)^{-1} X^T \vec{y}$$

$$= \left(\begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 \cdots 1 & 0 \cdots & 0 \\ 0 & \cdots & 0 & 1 \cdots 1 & 1 \cdots & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \end{bmatrix} \right)^{-1} X^T \vec{y}$$

*Simplification

$$= \begin{bmatrix} \frac{1}{nA} & 0 & 0 \\ 0 & \frac{1}{nB} & 0 \\ 0 & 0 & \frac{1}{nC} \end{bmatrix} \begin{bmatrix} \sum x_i = A\, y_i \\ \sum x_i - B\, y_i \\ \sum x_i = C\, y_c \end{bmatrix} = \underline{\begin{bmatrix} \bar{y}A \\ \bar{y}B \\ \bar{y}C \end{bmatrix} = \vec{b}}$$

(j) [harder] [MA] Prove that the OLS model always has $R^2 \in [0, 1]$.