MATH 390.4 / 650.2 Spring 2018 Homework #2t

Joseph Peltroche

Tuesday 6th March, 2018

Problem 1

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?

Solution. For the support vector machine algorithm,

$$\mathcal{H} = \left\{\mathbb{1}_{\vec{w}\cdot\vec{x}+b>0} : \vec{w} \in \mathbb{R}^P, b \in \mathbb{R}\right\}$$

$\mathcal{H}$ is different from the $\mathcal{H}$ used for the perceptron learning algorithm due its parameters. It takes in two parameters $\vec{w} \in \mathbb{R}^p$ and $b \in \mathbb{R}$ as opposed to the parameter $\vec{w} \in \mathbb{R}^{p+1}$ used in the perceptron learning algorithm. □

(b) [E.C.] Why is the SVM better than the perceptron? A non-technical discussion that makes sense is fine. Write it on a separate page

Solution. The SVM is better than the perceptron because the perceptron cannot find a solution a case that is not linearly separable, it is restricted to deal with only linearly separable situtations. The SVM can go beyond the perceptron by dealing with these case and actually find a best fit line with some errors. In the linearly separable case, the perceptron will also just spit out random lines within the margin, but the SVM will spit out the best line that is directly inbetween the margin. □

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

Solution. Analogous to the figure, we begin by coercing the $\delta = 1$. Instead of 0's we will replace them with -1's. Focusing on the border between 1's and the margin, our equation reads:

$$\begin{aligned} \vec{w} \cdot \vec{x} + b + 1 &= 0 \\ \vec{w} \cdot \vec{x} + b &= -1 \end{aligned}$$

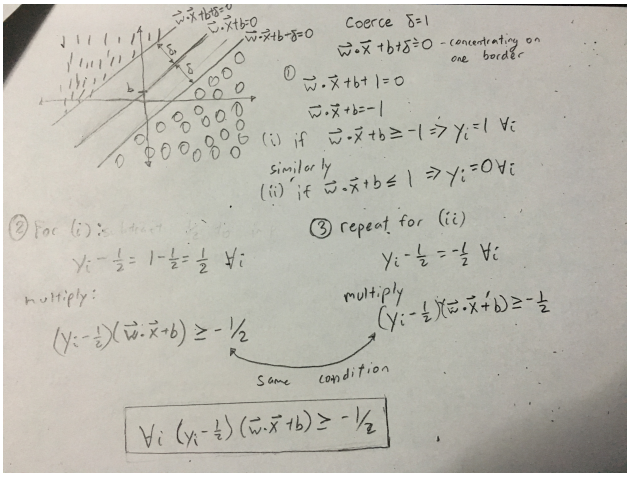


Figure 1: This depicts the method I used to solve for the condition of linear separability. The method if analogous to the one I will use for $\mathcal{Y} = \{-1, 1\}$

implying if $\vec{w} \cdot \vec{x} + b \geq -1 \Rightarrow y_i = 1 \forall i$ for all lines over or on that border. Similar, we can do the same for border of the margin and 0's. Now the equation will be $\vec{w} \cdot \vec{x} + b - 1 = 0$. We get the implication: if $\vec{w} \cdot \vec{x} + b \leq 1 \Rightarrow y_i = -1 \forall i$.

We then multiply the first implication by y_i on the left, and 1 on right

$$y_i(\vec{w} \cdot \vec{x} + b) \geq (1)(-1)$$

Similarly, we do the same to the last implication. Notice this time the signs will change bc we are multiplying the right by a negative number -1.

$$y_i(\vec{w} \cdot \vec{x} + b) \geq (1)(-1)$$

And the two equations match! Therefore our cost function for the linear separable case is

$$\forall i, y_i(\vec{w} \cdot \vec{x} + b) \geq (-1)$$

□

(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.

Solution. Since we are considering the soft margin solution, there exists points some hinge loss, and some points will not meet the condition of linear separability. In this case, $y_i(\vec{w} \cdot \vec{x} + b) < -1$ or equivalently $y_i(\vec{w} \cdot \vec{x} + b) = -1 + d$. Plugging this into our derived equation for linear separability:

$$\cancel{-1} + d \geq \cancel{-1}$$

or

$$d \geq 0$$

So we need to take this value whenever there is a hinge loss and return a 0 whenever there isn't. Thus

$$H_i = max\left\{0, -1 - y_i(\vec{w} \cdot \vec{x} + b)\right\}$$

This term, summed up over n, will return the average hinge loss. The cost function will also incoporate a term that maximizing the margin, which will establish a tradeoff between the amount of errors and best line of separation.

$$\underbrace{\frac{1}{n}\sum_{i=1}^{n} H_i}_{\text{average hinge loss}} + \underbrace{\lambda||\vec{w}||^2}_{maximizing\ the\ margin}$$

where λ is the hyperparameter. □

Problem 2

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

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

Solution. The KNN algorithm locates the closest $\vec{x}_i \in \mathcal{D}$ to our new data $\vec{x}^*$ by means of some meteric (distance function). Once the k-nearest neighbors are found, then

$$\hat{y} = mode[y(1), \ldots, y(k)]$$

which takes the mode (most frequently occuring member) of the responses of the k closest $\vec{x}_i's$. In this algorithm k is a hyperparameter because it will be predefined constant we chose to tune our algorithm □

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

Solution. The $\mathcal{H}$ is dependent on the metric between our new features and the observations of our training set and dependent on the k. Since $\mathcal{H}$ is defined as the set of all candidate functions, within the set there are functions that calculate distances between the new observation it intakes and the old feature (up to the kth)from the training data. This goes on from 1→n for the nth dimension of the training data. The functions will calculate the distance between the new observation point and old ones for the kth point. It will return the closest y's that are within the region calculated by the function in $\mathcal{H}$ and spit out the mode of the y's for each new feature. □

(c) [difficult] 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).

Solution. With $k = 1$, there will only be a single nearest neighbor. The prediction $\hat{y}$ would then always be equal to y_i since there is only one y_1, $y_i = \hat{y}$. Therefore the difference between the two would be nothing, and there would be zero error. This is not a good estimate of future error; by limiting our algorithm to $k = 1$, we can only compare the nearest neighbor to itself. Once new data is introduced, there will be distance between the new points and the previous nearest neighbor which will introduce error. □

Problem 3

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 $\mathcal{Y}$?

Solution.

$$\mathbb{D} = \left\{ \begin{bmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{n1} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \right\}$$

The X matrix in the training set has been reduced to a vector because $p = 1$. Both $\mathcal{X}, \mathcal{Y} \in \mathbb{R}$. □

(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.

Proof. To say that the point $< \bar{x}, \bar{y} >$ is on the OLS line is to say that $g(\bar{x}) = b_0 + b_1\bar{x} = \bar{y}$. We know that $b_1 = r\frac{S_y}{S_x}$ and $\bar{y} - r\frac{S_y}{S_x}\bar{x}$. Plugging into $g(\bar{x})$:

$$g(\bar{x}) = \bar{y} - r\frac{\cancel{S_y}}{\cancel{S_x}}\bar{x} + r\frac{\cancel{S_y}}{\cancel{S_x}}\bar{x}$$
$$g(\bar{x}) = \bar{y}$$

Hence the point $< \bar{x}, \bar{y} >$ is on the 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 $\bar{y}$.

Proof. To say the average prediction $\hat{y}_i$ for $x_i \in \mathbb{D}$ is $\bar{y}$ is to say $\frac{1}{n}\sum_i^n \hat{y}_i = \bar{y}$. Recall $b_1 = r\frac{S_y}{S_x}$ and $b_0 = \bar{y} - r\frac{S_y}{S_x}\bar{x}$. Plugging in for the expression,

$$\frac{1}{n}\sum_i^n \hat{y}_i = \frac{1}{n}\sum_n^i (b_0 + b_1 x_i)$$

$$
\begin{aligned}
&= \frac{1}{n}\sum_{i}^{n}\left(\bar{y} - r\frac{S_y}{S_x}\bar{x} + r\frac{S_y}{S_x}x_i\right) \\
&= \frac{1}{n}\left(\cancel{n}\bar{y} - \cancel{n}r\frac{S_y}{S_x}\bar{x} + r\frac{S_y}{S_x}\sum_{i}^{n}x_i\right) \\
&= \bar{y} - r\frac{S_y}{S_x}\bar{x} + \frac{1}{\cancel{n}}r\frac{S_y}{S_x}\bar{x}\cancel{n} \\
&= \bar{y} - r\cancel{\frac{S_y}{S_x}}\bar{x} + r\cancel{\frac{S_y}{S_x}}\bar{x} \\
&= \bar{y}
\end{aligned}
$$

the average prediction is equal to $\bar{y}$. □

(d) [harder] Consider the line fit using OLS. Prove that the average residual e_i computed from all predictions for $x_i \in \mathbb{D}$ and its true response value y_i is 0.

Proof. The average residuals can be expressed was $\frac{1}{n}\sum_i^n e_i = \frac{1}{n}\sum_i^n \hat{y}_i - y_i$. Since we are considering the OLS method, then we can use the coefficents $b_0 = \bar{y} - r\frac{S_y}{S_x}\bar{x}$ and $b_1 = r\frac{S_y}{S_x}$. Plugging into our expression:

$$
\begin{aligned}
\frac{1}{n}\sum_{i}^{n}\hat{y}_i - y_i &= \frac{1}{n}\sum_{i}^{n} b_0 - b_1 x_i - y_i \\
&= \frac{1}{n}\sum_{i}^{n}\left(\bar{y} - r\frac{S_y}{S_x}\bar{x} + r\frac{S_y}{S_x}x_i\right) - \frac{1}{n}\sum_{i}^{n} y_i \\
&= \frac{1}{n}\left(\cancel{n}\bar{y} - \cancel{n}r\frac{S_y}{S_x}\bar{x} + r\frac{S_y}{S_x}\sum_{i}^{n}x_i\right) - \bar{y} \\
&= \cancel{\bar{y}} - r\cancel{\frac{S_y}{S_x}}\bar{x} + r\cancel{\frac{S_y}{S_x}}\bar{x} - \cancel{\bar{y}} \\
&= 0
\end{aligned}
$$

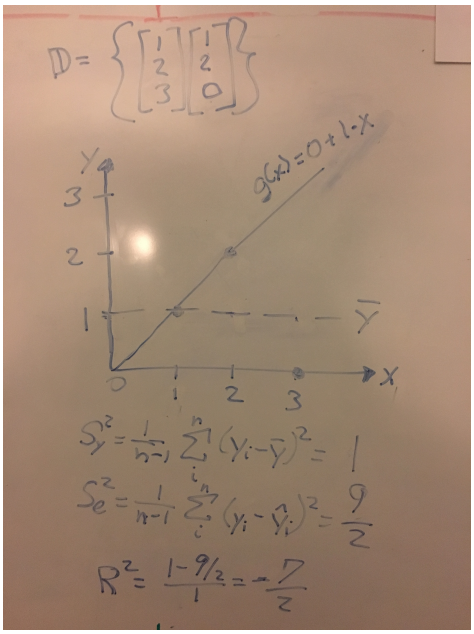
the average residual e_i computed for all predictions is 0. □

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

Solution. R^2 is interpreted to be the "proportion of the variance explained by the model" and compares the variance to that of null model. The sample variance of error S_e might be close to 0 and/or the sample variance of y might be huge, implying that R^2 is close to 1, which implies a "good" model. The greater R^2, the better fit line you have. Yet, even if $R^2 \approx 1$, the residuals themselves could be huge and our predictions could be far off from the actual results. The RMSE gives us an actual value corresponding

to our data that indicates just how off our predictions are from the actual data. In this way the RMSE is a better indicator than R^2. □

(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 $R^2 < 0$. Hint: do not use the OLS line. Hint: draw a picture!



(g) [E.C.] Prove that the OLS line always has $R^2 \in [0, 1]$ on a separate paper

Proof. R^2 is defined to be $R^2 = \frac{S_y^2 - S_e^2}{S_y^2}$, where S_e is the sample deviation of the errors of the OLS method and S_y is the sample deviation of the null model.

$$S_y = \frac{1}{n-1}\sum_i^n (y_i - \bar{y})^2$$

and

$$S_e = \frac{1}{n-1}\sum_i^n (e_i - \cancelto{0}{\bar{e}})^2 = \frac{1}{n-1}\sum_i^n e_i^2 = \frac{1}{n-1}\sum_i^n (y_i - \bar{y})^2$$

Recall $\bar{e} = 0$ from the proof to problem 3.d. R^2 can also be expressed as

$$R^2 = \frac{S_y^2 - S_e^2}{S_y^2} = 1 - \frac{S_e^2}{S_y^2}$$

If there is no sample deviation of the errors, then $S_e = 0$ and $R^2 = 1$. If $S_e > 0$ (because the sum of squares can only be zero or positive), then we want to show

$$\frac{S_e^2}{S_y^2} \leq 1$$

or $S_e^2 \leq S_y^2$, i.e.

$$\frac{1}{n-1}\sum_i^n (y_i - \hat{y}_i)^2 \leq \frac{1}{n-1}\sum_i^n (y_i - \bar{y})^2$$

This will show $R^2 \in [0, 1]$ by llimiting $0 < \frac{S_e^2}{S_y^2} < 1$. Assume that it isn't, so

$$\frac{1}{n-1}\sum_i^n (y_i - \hat{y}_i)^2 > \frac{1}{n-1}\sum_i^n (y_i - \bar{y})^2$$

Canceling out the fraction $\frac{1}{n-1}$ on both sides, we get:

$$\sum_i^n e_i^2 > \sum_i^n (y_i - \bar{y})^2$$

where $e_i = y_i - \hat{y}_i$ and $y_i - \bar{y}$ is the i-th error from the null model. On the contrary, we derived the OLS model by minimizing the SSE (sum of squared errors) and obtaining a best fit line that minimizes the residuals. There, the OLS model (Ordinary Least Squares) cannot be greater than the of sum squared errors for the null model since it already is the minimum. Therefore

$$\sum_i^n e_i^2 \leq \sum_i^n (y_i - \bar{y})^2$$

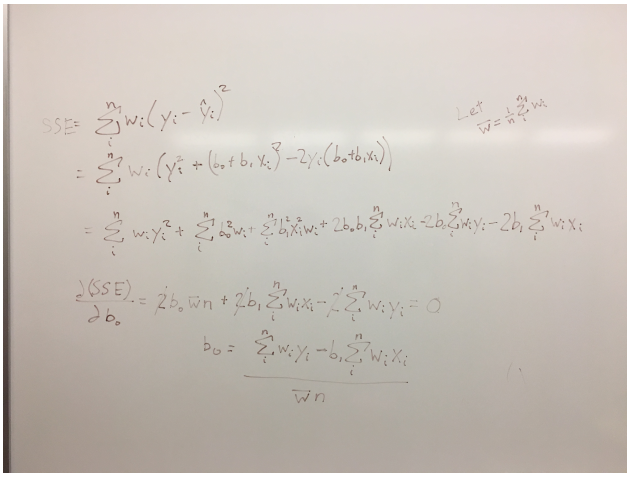
and

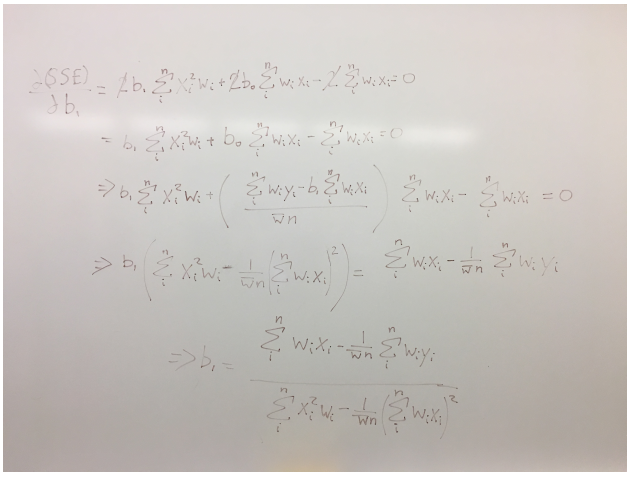
$$\begin{aligned} 0 \leq S_e^2 &\leq S_y^2 \\ \Rightarrow 0 \leq \frac{S_e^2}{S_y^2} &\leq 1 \\ \Rightarrow 0 \geq -\frac{S_e^2}{S_y^2} &\geq -1 \end{aligned}$$

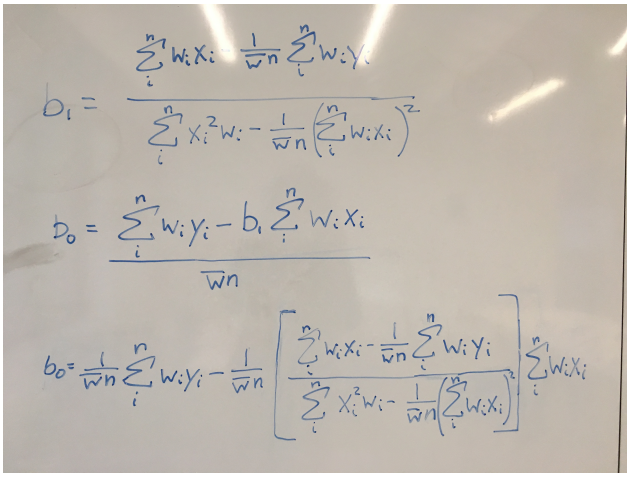
$$\Rightarrow 1 \geq 1 - \frac{S_e^2}{S_y^2} \geq 0 \qquad \text{(Adding 1 to all sides)}$$
$$\Rightarrow 1 \geq R^2 \geq 0$$

□

(h) [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.







Note: I included the images of my work due to all of the ugly sums in this problem

(i) [E.C.] 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?

Solution. The estimates from the ugly sums above were derived in a similar fashion to the estimates from the OLS. The major difference is the new "weighted" x_i's and y_i's. Now that they are weighted, there are some apparent differences, but for the most part both estimates resemble each other (before actually simplifying for the ugly sums in terms of r, S_x, S_y). This makes sense as the sums themselves include the i-th weight, weighing down and affecting our the x_i's and y_i's, but keeping the same structure of the OLS b_0 and b_1 expressions. □