MATH 390.4 / 650.2 Spring 2018 Homework #2t

Darshan Patel

Sunday 4th 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?

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

This is the same $\mathcal{H}$ used for $\mathcal{A}$ = perceptron learning algorithm except it has gotten reparameterized.

(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.
See page 6.

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

Let $\vec{w}$ be a normal vector, perpendicular to $\vec{x}$, such that $\vec{w}\cdot\vec{x} = 0$. Note $||\vec{w}||$ represents the length of the vector and $\vec{w}_0$ is a normalized vector defined in the same direction as $\vec{w}$ but with length 1. Let $\vec{l}$ be the vector from the origin to the line $\vec{w}\cdot\vec{x} + b = 0$ (perpendicular to it) which will be $\vec{l} = \alpha\vec{w}_0$. In fact, $\vec{l}$ is on the line $\vec{w}\cdot\vec{x} - b = 0$. Then

$$\begin{aligned}
\vec{w}\cdot\vec{l} - b &= 0 \\
\vec{w}\cdot\alpha\vec{w}_0 - b &= 0 \\
\vec{w}\cdot\alpha\frac{\vec{w}}{||\vec{w}||} - b &= 0 \\
\alpha\frac{||\vec{w}||^2}{||\vec{w}||} - b &= 0 \\
\alpha &= \frac{b}{||\vec{w}||}
\end{aligned}$$

To find the best line, note that the distance between the best line and the upper line is $\frac{b+\delta}{||\vec{w}||}$ and the distance between the best line and the lower line is $\frac{b-\delta}{||\vec{w}||}$. Here δ is the

difference in y between 2 lines. Then the distance between the upper and lower lines is

$$\frac{b+\delta}{||\vec{w}||} - \frac{b-\delta}{||\vec{w}||} = \frac{2\delta}{||\vec{w}||}$$

Let $\delta = 1$, then there's a unique solution to $\vec{w}\cdot\vec{x} - (b+\delta) = \vec{w}\cdot\vec{x} - b - 1 = 0$. Constrain all $y = 1$s to be above the upper line and all $y = -1$ to be below the lower line. Then

$$\vec{w}\cdot\vec{x} - b - 1 \geq 0 \rightarrow \vec{w}\cdot\vec{x} - b \geq 1$$

This is true for all $y_i = 1$. Multiply both sides by $y_i - \frac{1}{2}$.

$$(y_i - \frac{1}{2})(\vec{w}\cdot\vec{x}_i - b) \geq y_i - \frac{1}{2}$$

Since $y_i = 1$,

$$(y_i - \frac{1}{2})(\vec{w}\cdot\vec{x}_i - b) \geq 1 - \frac{1}{2} = \frac{1}{2}$$

For the other case, $y_i = 0$ and so

$$\vec{w}\cdot\vec{x}_i - (b-1) \geq 0 \rightarrow \vec{w}\cdot\vec{x}_i - b \geq -1$$

Multiply both sides by $y_i - \frac{1}{2}$ and substitute 0 for y_i

$$(y_i - \frac{1}{2})(\vec{w}\cdot\vec{x}_i - b) \geq -(y_i - \frac{1}{2}) = \frac{1}{2}$$

Therefore the condition of perfect separability is : if the point is on the line, it equals $\frac{1}{2}$ and if not, it is greater than $\frac{1}{2}$.

(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.

Let the margin be $|\alpha| = \frac{1}{||\vec{w}||}$. To maximize it, we have to minimize $||\vec{w}||$. But we can't minimize both the original cost function and $||\vec{w}||$. Therefore if we make our cost function

$$\frac{1}{n}\sum \max\left\{0, \frac{1}{2} - (y_i - \frac{1}{2})(\vec{w}\cdot\vec{x}_i - b)\right\} + \lambda\,||\vec{w}||^2$$

we set an upper bound for what the maximum margin can be, λ, a predefined 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"?

The algorithm works by computing a type of distance between 2 points, one from the actual data set, and one from a training data set. It calculates distances from the entire set and then pick out the point where the least distance was measured k times and return the y value at that point. k is a "hyperparameter" because it is a predefined constant determined by the user to tell the algorithm how many closest neighbors to pick.

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

Assuming $\mathcal{A} = \text{KNN}$, the input $\mathcal{H}$ is $\{x : x \in \mathbb{R}\}$. It is a collection of constants that can predict a single value y. The KNN algorithm returns the value at the point where distant is most minimal therefore we should except $\mathcal{H}$ to be a collection of constants from $\mathbb{R}$.

(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).

There should be zero error when predicting on $\mathbb{D}$ with $k = 1$ because the training data will be used on itself to find the nearest neighbor. When doing so, a point's nearest neighbor in will always be the point itself. Therefore no error should be made. This is not a good estimate of future error when new data comes in because it is not certain whether the data will fit in the range of current data.

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 $\mathcal{Y}$?

$$\mathcal{D} = \left\{ \begin{bmatrix} x_{11} \\ X_{21} \\ \dots \\ X_{n1} \end{bmatrix}, \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} \right\}$$

$\mathcal{X}$ is a collection of input points and $\mathcal{Y}$ is the collection of output points.

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

Let the fitted line be $y = b_0 + b_1 x$ where $b_0 = \bar{y} - r\frac{s_y}{s_x}\bar{x}$ and $b_1 = r\frac{s_y}{s_x}$. Then if we solve for y where $x = \bar{x}$,

$$\begin{aligned} y &= b_0 + b_1 x \\ &= \left(\bar{y} - r\frac{s_y}{s_x}\bar{x}\right) + r\frac{s_y}{s_x}x \\ &= \bar{y} - r\frac{s_y}{s_x}\bar{x} + r\frac{s_y}{s_x}\bar{x} \\ &= \bar{y} \end{aligned}$$

Clearly $\langle \bar{x}, \bar{y} \rangle$ is on the 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 $\bar{y}$.

$$\begin{aligned} y &= w_0 + w_1 x \\ \frac{\sum_i^n y_i}{n} &= \frac{\sum_i^n w_0 + w_1 x_i}{n} \\ \bar{y} &= \frac{n w_0}{n} + \frac{n w_1 \bar{x}}{n} \\ \bar{y} &= w_0 + w_1 \bar{x} \end{aligned}$$

(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.

$$\begin{aligned} e_i &= y_i - (b_0 + b_1 x_i) \\ \frac{1}{n}\sum_i^n e_i &= \frac{1}{n}\sum_i^n y_i - (b_0 + b_1 x_i) \\ &= \frac{1}{n}\sum_i^n y_i - y_i \\ &= 0 \end{aligned}$$

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

The RMSE is usually a better indicator of predictive performance than R^2 because it acts as a standard deviation. It explains the average deviation of the estimates from the observed values. Furthermore, it has the same units as y and can be interpreted in English. R^2, however, is unit-less and between 0 and 1. It only explains, in percent, how much of the variance was captured by the estimate. The RMSE goes a step further and instead explains how bad the errors were, from 0 to "∞".

(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!

$$R^2 = \frac{SSE_0 - SSE}{SSE_0} = 1 - \frac{SSE}{SSE_0}$$

According to this equation, we can attain a negative R^2 value when $\frac{SSE}{SSE_0} > 1$, or $SSE > SSE_0$. What this means is that the sum of squared errors from a trend line created is greater than the one created from the null model $\bar{y}$. An instance where this happens is where we have a positively correlated dataset and a negative trend line is

constructed. Say

$$\mathcal{D} = \{(1,1),(4,10)\}$$

This is a positively correlated dataset where $\bar{y} = 9$. A perfect model for this dataset would be $g(x) = -2 + 3x$ where $R^2 = 1$. A model for where R^2 is less than 0, or did worse that a horizontal line at $\bar{y}$, is $g(x) = 6 - 3x$.

(g) [E.C.] Prove that the OLS line always has $R^2 \in [0,1]$ on a separate page.
See page 6.

(h) [difficult] You are given $\mathbb{D}$ with n training points $\langle x_i, y_i \rangle$ 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.)

$$\begin{aligned}
SSE &= \sum w_i(y_i - \hat{y}_i)^2 = \sum w_i(y_i - (w_0 + w_1 x_i))^2 \\
&= \sum w_i(y_i^2 + w_0^2 + w_1^2 x_i^2 - 2y_i w_0 - 2y_i w_1 x_i + 2w_0 w_1 x_i) \\
&= \sum w_i y_i^2 + w_0^2 \sum w_i + w_1^2 \sum w_i x_i^2 - 2w_0 \sum w_i y_i \\
&\quad - 2w_1 \sum w_i x_i y_i + 2w_0 w_1 \sum w_i x_i \\
\frac{\partial SSE}{\partial w_0} &= 2w_0 \sum w_i - 2\sum w_i y_i + 2w_1 \sum w_i x_i \overset{\text{set to}}{=} 0 \\
0 &= w_0 \sum w_i - \sum w_i y_i + w_1 \sum x_i \\
w_0 \sum w_i &= \sum w_i y_i - w_1 \sum x_i \\
b_0 = w_0 &= \frac{\sum w_i y_i - w_1 \sum x_i}{\sum w_i} \\
\frac{\partial SSE}{\partial w_1} &= 2w_1 \sum w_i x_i^2 - 2\sum w_i x_i y_i + 2w_0 \sum w_i x_i \overset{\text{set to}}{=} 0 \\
w_1 \sum w_i x_i^2 &= \sum w_i x_i y_i - w_0 \sum w_i x_i \\
b_1 = w_1 &= \frac{\sum w_i x_i y_i - w_0 \sum w_i x_i}{\sum w_i x_i^2}
\end{aligned}$$

(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?
See page 7.

Problem 4

Extra Credit Problems

(a) [E.C.] Why is the SVM better than the perceptron? A non-technical discussion that makes sense is fine.
The SVM is better than the perceptron algorithm. The SVM and perceptron both assume linear separability. But it is only the SVM algorithm that takes the data values in close consideration. On one hand, the initial weights in the perceptron algorithm are user-defined. When the algorithm runs, it'll find some line that separates binary values. Depending on the choice of the initial weights, it can give very different outcomes. Furthermore, it can give results that don't seem to evenly divide the areas of the two values. This is different from the SVM algorithm. Here a line is drawn such that it separates two outcomes depending on the distance between the outcomes. It takes into account of all edge points and finds the midpoint of it for where the line should go through.

(b) [E.C.] Prove that the OLS line always has $R^2 \in [0, 1]$
Note that the OLS line has the form The formula for R^2 is as follows:

$$R^2 = \frac{SSE_0 - SSE}{SSE_0} = \frac{s_y^2 - s_e^2}{s_y^2}$$

It happens to be that $s_y^2 = \frac{1}{n-1}\sum(y_i - \bar{y})^2$ and $s_e^2 = \frac{1}{n-1}\sum e_i^2$. This means

$$R^2 = \frac{\frac{1}{n-1}\sum(y_i - \bar{y})^2 - \frac{1}{n-1}\sum e_i^2}{\frac{1}{n-1}\sum(y_i - \bar{y})^2} = \sum \frac{(y_i - \bar{y})^2 - e_i^2}{(y_i - \bar{y})^2} = 1 - \sum \frac{e_i^2}{(y_i - \bar{y})^2}$$

We can deduce that $\sum \frac{e_i^2}{(y_i - \bar{y})^2} \in [0, 1]$ by the following reasoning. If we attain a line that is close to $\bar{y}$, then

$$\lim_{y_i \to \bar{y}} \frac{e_i^2}{(y_i - \bar{y})^2} = \lim_{y_i \to \bar{y}} \frac{(y_i - \bar{y})^2}{(y_i - \bar{y})^2} = 1$$

and so $R^2 = 1 - 1 = 0$. This is the lower bound of R^2. Now suppose we achieve the best possible line. This means that $e_i \approx 0$. Therefore

$$\frac{e_i^2}{(y_i - \bar{y})^2} \to \frac{0}{(y_i - \bar{y})^2} = 0$$

and so $R^2 = 1 - 0 = 1$. This is the upper bound of R^2. Therefore the OLS line always has $R^2 \in [0, 1]$. $\square$

(c) [E.C.] Interpret the ugly sums in the b_0 and b_1 you derived in $3h$ 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?
For the unweighed case,

$$b_0 = \bar{y} - r\frac{s_y}{s_x}\bar{x} = \bar{y} - b_1\bar{x} \text{ and } b_1 = r\frac{s_y}{s_x} = \frac{s_{xy}}{s_x s_y}\frac{s_y}{s_x} = \frac{s_{xy}}{s_x^2}$$

For the weighed case,

$$b_0 = \frac{\sum w_i y_i - w_1 \sum x_i}{\sum w_i} \text{ and } b_1 = \frac{\sum w_i x_i y_i - w_0 \sum w_i x_i}{\sum w_i x_i^2}$$

In both cases, we see that the b_0 terms are nearly the same. The only alteration is that in the weighed case, both terms in the numerator are affected by a weight and then divided by the sum of the weight. The same goes for the b_1 terms. Both equations are of the same standard deviation from. In the weighed case, each summation is altered by a weight. The terms in this derivation makes sense in the weighed least squares alteration because we want the weights to affect the error for each of the $\langle x_i, y_i \rangle$ points.