Suggestions, PS 11

9.5. (a) What is the average number of points per region? Show in this case each $\widehat{y}_i$ is the average of approximately N/m points y_i , i.e.,

$$\widehat{y}_i = \frac{m}{N} \sum_{j \in R(i)} y_j,$$

where $R(i)$ is the region (out of m) in which the point y_i is defined. Thus show

$$\text{cov}(y_i, \widehat{y}_i) = \frac{m}{N} \sum_{j \in R(i)} \text{cov}(y_i, y_j) = \frac{m}{N} \text{cov}(y_i, y_i).$$

Hence show the df are

$$\frac{1}{\sigma^2} \sum_i \text{cov}(y_i, \widehat{y}_i) = \frac{1}{\sigma^2} N \frac{m}{N} \sigma^2 = m.$$

(c) Here for each choice $m = 1, 5,$ or $10,$ you should generate 10 regression trees, each based on new values of y_i, (keeping the same $\mathbf{x}_i$). This will for each point $\mathbf{x}_i$ give you 10 values of y_i and 10 values of $\widehat{y}_i$, so you can estimate their covariance for each i, which can then average. One package you could try that lets you control the number of leaf nodes (rectangles) is the tree package in R (https://www.rdocumentation.org/packages/tree/versions/1.0-39), and the program prune.tree.

(d) Are the degrees of freedom comparable in (a) and (c)? If not, what does this show about our approximate argument in (a)? Where are the additional degress of freedom in (c) coming from? That is, what freedom is there in the choice of the regression function (from the tree) beside just the values of the regression function at each node (in each rectangle)?

(e) Imagine we divide the entire domain R where $\mathbf{x}$ values are drawn into m equal sized parts $R_1, ..., R_m$ and fix these (instead of determining them with the regression tree). In that case show we would then regress inside each R_i independently, getting a linear method. Specifically, show

$$\widehat{\mathbf{y}} = \mathbf{S}\mathbf{y} \,,$$

where $\mathbf{S}$ is the matrix which averages the values $y_i,$ so

$$(\mathbf{S}\mathbf{y})_i = \frac{1}{N(i)} \sum_{\mathbf{x}_j \in R(\mathbf{x}_i)} y_j \,,$$

where $R(\mathbf{x}_i)$ is the region containing $\mathbf{x}_i$, and $N(i)$ is the number of data points in it. Deduce the entries of matrix $\mathbf{S}$ - specifically, show

$$S_{ij} = \begin{cases} \frac{1}{N(i)} & \text{if } \mathbf{x}_j \in R(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases} .$$

You can then compute the trace (as in (a)) to be

$$\text{tr } \mathbf{S} = \sum_{i=1}^{N} \frac{1}{N(i)} = m .$$

Note for R_j, the sum over all i with $\mathbf{x}_i \in R_j$ equals 1, and there are m such pieces in the sum.

For a more realistic estimate, you can think about a relationship like $\widehat{\mathbf{y}} = \mathbf{S}\mathbf{y}$ in terms of components. For example, this would say that if $\mathbf{S} = (s_{ij})$ are the entries of $\mathbf{S}$, then we expect $\widehat{y}_1 = \sum_j s_{1j} y_j = \mathbf{s}_1 \mathbf{y}$, where $\mathbf{s_1} = [s_{11}\, s_{12} \ldots\, s_{1N}]$ is the first row of $\mathbf{S}$ (note that $\mathbf{s}_1$ is a row vector and $\mathbf{y}$ is a column). If you fix the data points $\mathbf{x}_i$ and vary the y_i over say 10 datasets, can you try to do a regression of $\widehat{y}_1$ on the y_i to find estimates for the s_{1j}? This will give you the first row $\widehat{\mathbf{s}}_1$ of the estimate for $\mathbf{S}$. You can do subsequent rows similarly, to get an estimate $\widehat{\mathbf{S}}$ for the matrix $\mathbf{S}$ (just for the fixed set of points $\{\mathbf{x}_i\}_{i=1}^N$; remember we are fixing these and varying only the y_i). What is tr $\widehat{\mathbf{S}}$? Is it closer to the answer in (a) or in (c)?

Actually, over 10 datasets there are not enough examples to fit the above 100 parameters in $\widehat{\mathbf{s}}_1$. To do this you should instead try for 1000 repetitions instead of just 10, and have an algorithm that can perform this regression to find $\mathbf{s}_1$ (and then the rest of the rows of $\mathbf{S}$) in a reasonable amount of computer time.

10.1 To minimize (11) take the derivative and set it to 0. Defining

$$A = \sum_{i=1}^{N} w_i^{(m)} I_{\{y_i \neq G(x_i)\}} , \; C = \sum_{i=1}^{N} w_i^{(m)} ,$$

(minimize $(e^{\beta} - e^{-\beta})A + e^{-\beta}C$. You can set the derivative to 0 - show you get

$$(e^{2\beta} + 1)A = C$$

$$e^{2\beta} = \frac{C}{A} - 1.$$

10.2 Fixing $\mathbf{x}$, show we must choose $f(\mathbf{x})$ to minimize

$$E_{y|\mathbf{x}}(e^{-yf(\mathbf{x})}) = e^{-f(\mathbf{x})} P(y = 1) + e^{f(x)} P(y = -1) .$$

Defining $A = e^{f(x)}$, minimize (with respect to A) $A^{-1}P(y=1) + AP(y=-1)$. Taking the derivative show that:

$$-A^{-2}P(y=1) + P(y=-1) = 0.$$

Now solve for A.

10.7 Show

$$\widehat{\gamma}_{jm} = \underset{\gamma_{jm}}{\text{argmin}} \sum_{x_i \in R_{jm}} e^{-y_i(f_{m-1}(x_i)+\gamma_{jm})}$$

$$= \underset{\gamma}{\text{argmin}} \sum_{x_i \in R_{jm}} e^{-y_i \gamma_{jm}} w_i^{(m)}$$

Show that finding $\widehat{\gamma}_{jm}$ is exactly the same as the derivation of β in class (see also equation (10.12), with the role of $\beta G_m(\mathbf{x}_i)$ played by γ. Thus verify that if we set $G_m(\mathbf{x}_i) = 1$ for all i, then we can set $\gamma = \beta$ in the formula $\beta = \frac{1}{2}\frac{1-\text{err}_m}{\text{err}_m}$. Show that now

$$\text{err}_m = \frac{\sum_{i=1}^{N} w_i^{(m)} I(y_i \neq 1)}{\sum_{i=1}^{N} w_i^{(m)}},$$

and so

$$\gamma = \frac{1}{2} \frac{\sum_{i=1}^{N} w_i^{(m)} I(y_i = 1)}{\sum_{i=1}^{N} w_i^{(m)} I(y_i \neq 1)}.$$