STAT 745 – Fall 2014
Individual Assignment 1

Doug Raffle

October 1, 2014

1 Graph-Based Splines

1.a Derivation

Show that the solution to:

$$\min_f (y-f)^T W (y-f) + f^T \Delta f \tag{1}$$

is equivalent to local regression of an intercept. In Assignment Three, we showed that this estimator is for an observation x_0:

$$\hat{f} = \frac{\sum_{i=1}^n K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^n K_\lambda(x_0, x_i)} \tag{2}$$

We start by differentiating Equation (1) with respect to f.

$$\frac{\partial}{\partial f} (y-f)^T W (y-f) + f^T \Delta f = -(y-f)^T (W + W^T) + f^T (\Delta + \Delta^T)$$

The derivative is set to zero and we solve for f.

$$\begin{aligned}
0 &= -(y-f)^T (W + W^T) + f^T (\Delta + \Delta^T) \\
&= -(y^T - f^T)(W + W^T) + f^T (\Delta + \Delta^T) \\
&= (f^T - y^T)(W + W^T) + f^T (\Delta + \Delta^T) \\
&= f^T (W + W^T) - y^T (W + W^T) + f^T (\Delta + \Delta^T) \\
y^T (W + W^T) &= f^T (W + W^T) + f^T (\Delta + \Delta^T) \\
y^T (W + W^T) &= f^T (W + W^T + \Delta + \Delta^T) \\
y^T (W + W^T) &= f^T (W + \Delta + (W + \Delta)^T)
\end{aligned}$$

Since $\Delta = D - W$, it follows that $D = W + \Delta$.

$$y^T (W + W^T)(D + D^T)^{-1} = f^T$$

We can make use of the fact that W and D are both symmetric (i.e., $W = W^T$).

$$\begin{aligned}
y^T (2W)(2D)^{-1} &= f^T \\
y^T W D^{-1} &= f^T
\end{aligned}$$

Finally, we take the transpose of both sides for our final estimator:

$$\hat{f} = D^{-1} W y \tag{3}$$

To compare it to (2), we can re-write (3) for a single observation (where w_i^T is the i^{th} row of W).

$$\hat{f}_i = D_{ii}^{-1} w_i^T y = \frac{\sum_{j=1}^n w_{ij}^T y_j}{\sum_{j=1}^n w_{ij}^T} = \frac{\sum_{j=1}^n K_\lambda(x_i, x_j) y_j}{\sum_{j=1}^n K_\lambda(x_i, x_j)} \quad \blacksquare$$

Which is equivalent to Equation (2) if $x_0 = x_i$

1.b Application

The graph is read in as the matrix W, and D^{-1} is calculated from this. Papers topics are read in as the vector y so that Artificial Intelligence papers are coded as zeros and Machine Learning papers as ones. The matrix calcuations from (a) are used to create a vector of predicted probabilities. These predictions are rounded to the nearest integer to find the predicted paper topic.

```
> W <- as.matrix(read.table("cite.txt", header=TRUE))
> D.inv <- diag(as.vector(W %*% rep(1, nrow(W)))^(-1))
> y <- as.numeric(read.table("class.txt", header=TRUE)[,1]) - 1
> y.hat <- round(D.inv %*% W %*% y)
```

We can use the rate of misclassified topics as a measure of training error. It may also be of interest to examine the error rates within each topic.

```
> err.rate <- function(y, y.hat){
+     o.err <- length(which(y != y.hat))/ length(y) * 100
+     ml.err <- length(which(y == 1 & y.hat == 0))/length(which(y == 1)) * 100
+     ai.err <- length(which(y == 0 & y.hat == 1))/length(which(y == 0)) * 100
+     round(data.frame("Overall" = o.err, "ML" = ml.err, "AI" = ai.err), 2)
+ }
> err.rate(y, y.hat)

  Overall   ML   AI
1    3.87 3.25 4.38
```

There was an overall error rate of 3.87%. Artificial Intelligence papers were classified as ML at a slightly higher rate (4.38%) than Machine Learning papers were misclassified as AI (3.25%).

We can now compare these error rates to those of Least Squares Regression.

```
> lsr <- lm(y ~ as.matrix(read.table("title.txt", header=TRUE)))
> err.rate(y, round(predict(lsr)))

  Overall   ML    AI
1   17.86 24.5 12.11
```

Least Squares Regression has a much higher error rate than the graph-based approach, both in overall error and within the classes.

2 Smoothing Splines

2.a Derivation

Since we are interpolating between the knot points, the integral over the interval $[a, b]$ becomes a sum of the integrals between the knot points.

$$\int_a^b g''(x)h''(x)\,dx = \int_a^{x_1} g''(x)h''(x)\,dx + \sum_{i=1}^{N-1}\int_{x_i}^{x_{i+1}} g''(x)h''(x)\,dx + \int_{x_n}^{b} g''(x)h''(x)\,dx$$

Each of these regions can then be solved with integration by parts.

$$\begin{aligned}\int_a^b g''(x)h''(x)\,dx = &\left[g''(x)h'(x)|_a^{x_1} - \int_a^{x_1} g'''(x)h''(x)\,dx\right] \\ &+ \sum_{i=1}^{N-1}\left[g''(x)h'(x)|_{x_i}^{x_{i+1}} - \int_{x_i}^{x_{i+1}} g'''(x)h''(x)\,dx\right] \\ &+ \left[g''(x)h'(x)|_{x_N}^{b} - \int_{x_N}^{b} g'''(x)h''(x)\,dx\right]\end{aligned}$$

From here, we continue evaluating the integrals in each region. Note that the second and third derivatives in the boundary regions will be zero, since natural splines are linear here. Also, note that $h(x_i)$ is zero at any knot point, since both $g(x)$ and $g(\widetilde{x})$ must pass through the points.

$$\begin{aligned}\int_a^b g''(x)h''(x)\,dx =& [g''(x_1)h'(x_1) - g'''(x_1)h(x_1) - g''(a)h'(a) + g'''(a)h(a)] \\ &+ \left[\sum_{i=1}^{N-1} g''(x_{i+1})h'(x_{i+1}) - \sum_{i=1}^{N-1} g''(x_i)h'(x_i) - \sum_{i=1}^{N-1}\int_{x_i}^{x_{i+1}} g'''(x)h(x)\right] \\ &+ [g''(b)h'(b) + g'''(b)h(b) - g''(x_N)h'(x_N) - g'''(x_N)h(x_N)] \\ \\ =& g''(x_1)h'(x_1) - g''(x_N)h'(x_N) + g''(x_N)h'(x_N) - g''(x_1)h'(x_1) \\ &- g'''(x_1)h(x_1) + g'''(x_N)h(x_N) - \sum_{i=1}^{N-1}\int_{x_i}^{x_{i+1}} g'''(x)h(x) \\ =& -\sum_{i=1}^{N-1}\int_{x_i}^{x_{i+1}} g'''(x)h(x)\end{aligned}$$

Since these are cubic splines, the third derivative will be a constant in each region, which allows us to evaluate the remaining sum of integrals. However, using the same logic as above, $h(x_{i+1}) = h(x_i) = 0$.

$$\int_a^b g''(x)h''(x)\,dx = -\sum_{i=1}^{N-1} g'''(x_i^+)\left[h(x_{i+1}) - h(x_i)\right] = 0 \quad \blacksquare$$

2.b Curvature

Show that:

$$\int_a^b \widetilde{g}''(t)^2\,dt \geq \int_a^b g''(t)^2\,dt$$

Since $h(x) = \widetilde{g}(x) - g(x)$, it follows that $\widetilde{g} = h(x) + g(x)$. Using this fact:

$$\begin{aligned}\int_a^b \left(h''(x) + g''(x)\right)^2\,dt &\geq \int_a^b g''(t)^2\,dt \\ \int_a^b h''(x)^2 + 2g''(x)h''(x) + g''(x)^2\,dt &\geq \int_a^b g''(t)^2\,dt \\ \int_a^b h''(x)^2\,dt + \int_a^b 2g''(x)h''(x)\,dt + \int_a^b g''(x)^2\,dt &\geq \int_a^b g''(t)^2\,dt\end{aligned}$$

Using the result from (2.a), we arrive at:

$$\int_a^b h''(x)^2\,dt + \int_a^b g''(x)^2\,dt \geq \int_a^b g''(t)^2\,dt \quad \blacksquare$$

If $\widetilde{g}(x)$ is a natural cubic spline, $\widetilde{g}(x) = g(x)$, from which we know $h(x) = 0$. In this case, both sides of the expression are the same and the equality holds. Because of the squared term, $\int_a^b h''(t)^2\,dt \geq 0$, thus the inequality holds for non-zero values of $h(x)$.

2.c Application to Penalized Least Squares

An integral of the form in (2.b) penalizes functions with high curvature. As was shown above, a penalty of this form for a natural cubic spline is less than or equal to that of any differentiable function of x. Since the integral is smallest for a natural cubic spline, it follows that its product with λ will be similarly optimized in Penalized Least Squares.

3 Smoothing Splines

3.a Derive S_λ

$$\hat{f} = S_\lambda y = \min_f (y-f)^T(y-f) + \lambda f^T R f$$

We begin by differentiating with respect to f.

$$\frac{\partial}{\partial f}(y-f)^T(y-f) + \lambda f^T R f = -2(y-f)^T \lambda f^T (R + R^T)$$

Then we set this derivative equal to zero and solve for f.

$$\begin{aligned}
-2(y-f)^T \lambda f^T (R+R^T) &= 0 \\
2f^T - 2y^T + \lambda f^T (R+R^T) &= 0 \\
2f^T + 2\lambda f^T R &= 2y^T \\
f^T(I+\lambda R) &= y^T \\
\hat{f} = S_\lambda y &= (I+\lambda R)^{-1} y \\
S_\lambda &= (I+\lambda R)^{-1}
\end{aligned}$$

3.b Identify R Independent of λ

We know that:

$$\hat{f} = (I+\lambda R)^{-1} y = N(N^T N + \lambda \Omega_N)^{-1} N^T y$$

We can solve for R from here so that it is independent of λ.

$$\begin{aligned}
(I+\lambda R)^{-1} &= N(N^T N + \lambda \Omega_N)^{-1} N^T \\
I + \lambda R &= N^{-T}(N^T N + \lambda \Omega_N) N^{-1} \\
I + \lambda R &= N^{-T} N^T N N^{-1} + \lambda N^{-T} \Omega_N N^{-1} \\
I + \lambda R &= I + \lambda N^{-T} \Omega_N N^{-1} \\
R &= N^{-T} \Omega_N N^{-1}
\end{aligned}$$