STAT 745 – Fall 2014
Assignment 8

Group 1: Francene Cicia, Doug Raffle, Melissa Smith

November 3, 2014

1 Margin Loss Function

We want to show that for any margin loss function $L(yf)$, the minimizer of:

$$\min_f \sum_{i=1}^{n} L(y_i f(x_i))$$

is solved by $y_i = f(x_i)$ for all i.

We have $y, f \in \{-1, 1\}$. Any margin loss function $L(yf)$ can be written as an equivalent convex function of the difference, $\psi(y - f)$. Then the above minimization problem is equivalent to:

$$\min_f \sum_{i=1}^{n} \psi(y_i - f(x_i))$$

So we take the derivative and set it equal to zero:

$$\frac{\partial}{\partial f} : \sum_{i=1}^{n} \psi^{'}(y_i - f(x_i)) = 0$$

Given that this is a convex function, it is obvious that the minimum occurs when $f(x_i) = y_i$. This is inuitively reasonable. We want to penalize $f(x)$ when it misclassifies the observations, and we should incur the smallest penalty when we correctly classify the observation, i.e., $L(yf) < L(yf')$ where f is interpolation and f' is any other prediction where $f \neq y$.

2 Graph Based Classification

2.1 Positive Semi-Definite

Let $y_i \in \{-1, 1\}$. Denote W as the adjacency matrix of a graph. Define $\Delta = D - W$ where D is the row sum matrix of W.
(a) Show that Δ is positive semi-definite.

Let $\nu \neq \overrightarrow{0}$. Then,

$$W = \begin{bmatrix} W_{1,1} & W_{1,2} & \cdots & W_{1,n} \\ W_{2,1} & W_{2,2} & \cdots & W_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ W_{n,1} & W_{n,2} & \cdots & W_{n,n} \end{bmatrix} \quad D = \begin{bmatrix} \sum_{j=1}^{n} W_{1,j} & 0 & \cdots & 0 \\ 0 & \sum_{j=1}^{n} W_{2,j} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sum_{j=1}^{n} W_{n,j} \end{bmatrix} \quad \nu = \begin{bmatrix} \nu_1 \\ \vdots \\ \nu_n \end{bmatrix}$$

Using the fact that $\Delta = D - W$,

$$\begin{aligned} \nu^T \Delta \nu &= \nu^T (D - W) \nu \\ &= \left(\nu^T D - \nu^T W\right) \nu \\ &= \nu^T D \nu - \nu^T W \nu \end{aligned}$$

Expanding each of these terms, we get:

$$\begin{aligned}
\nu^T D\nu &= \left[\nu_1^2 \sum_{j=1}^n W_{1,j} + \nu_2^2 \sum_{j=1}^n W_{2,j} + \nu_3^2 \sum_{j=1}^n W_{3,j} + \cdots + \nu_n^2 \sum_{j=1}^n W_{n,j}\right] \\
&= \nu_1^2 \left(W_{1,1} + W_{1,2} + \cdots + W_{1,n}\right) + \nu_2^2 \left(W_{2,1} + W_{2,2} + \cdots + W_{2,n}\right) + \cdots \\
&\quad + \nu_{n-1}^2 \left(W_{n-1,1} + W_{n-1,2} + \cdots + W_{n-1,n}\right) + \nu_n^2 \left(W_{n,1} + W_{n,2} + \cdots + W_{n,n}\right)
\end{aligned}$$

$$\begin{aligned}
\nu^T W\nu &= \nu_1 \left(\nu_1 W_{1,1} + \nu_2 W_{2,1} + \cdots + \nu_n W_{n,1}\right) + \nu_2 \left(\nu_1 W_{1,2} + \nu_2 W_{2,2} + \cdots + \nu_n W_{n,2}\right) + \cdots \\
&\quad + \nu_n \left(\nu_1 W_{1,n} + \nu_2 W_{2,n} + \cdots + \nu_n W_{n,n}\right)
\end{aligned}$$

$$\begin{aligned}
\nu^T D\nu - \nu^T W\nu &= \nu_1^2 \left(W_{1,1} + W_{1,2} + \cdots + W_{1,n}\right) + \nu_2^2 \left(W_{2,1} + W_{2,2} + \cdots + W_{2,n}\right) + \cdots \\
&\quad + \nu_{n-1}^2 \left(W_{n-1,1} + W_{n-1,2} + \cdots + W_{n-1,n}\right) + \nu_n^2 \left(W_{n,1} + W_{n,2} + \cdots + W_{n,n}\right) \\
&\quad - \nu_1 \left(\nu_1 W_{1,1} - \nu_2 W_{2,1} - \cdots - \nu_n W_{n,1}\right) - \nu_2 \left(\nu_1 W_{1,2} - \nu_2 W_{2,2} - \cdots - \nu_n W_{n,2}\right) - \cdots \\
&\quad - \nu_n \left(\nu_1 W_{1,n} - \nu_2 W_{2,n} - \cdots - \nu_n W_{n,n}\right)
\end{aligned}$$

Notice that the terms of the form $\nu_i^2 W_{i,j}$ where $i = j$ cancel out when you expand. Also, W is a symmetric matrix, so $W_{i,j} = W_{j,i}$, thus when we simplify we get,

$$\begin{aligned}
\nu^T D\nu - \nu^T W\nu &= \left(\nu_1^2 - 2\nu_1\nu_2 + \nu_2^2\right) W_{1,2} + \left(\nu_1^2 - 2\nu_1\nu_3 + \nu_3^2\right) W_{1,3} + \cdots \\
&\quad + \left(\nu_1^2 - 2\nu_1\nu_n + \nu_n^2\right) W_{1,n} + \cdots + \left(\nu_2^2 - 2\nu_2\nu_3 + \nu_3^2\right) W_{2,3} \\
&\quad + \left(\nu_2^2 - 2\nu_2\nu_4 + \nu_4^2\right) W_{2,4} + \cdots + \left(\nu_2^2 - 2\nu_2\nu_n + \nu_n^2\right) W_{2,n} + \cdots \\
&\quad + \left(\nu_{n-1}^2 - 2\nu_{n-1}\nu_n + \nu_n^2\right) W_{n-1,n}
\end{aligned}$$

So because W is symmetric we have,

$$\begin{aligned}
2\left[\nu^T D\nu - \nu^T W\nu\right] &= 2\left[\left(\nu_1^2 - 2\nu_1\nu_2 + \nu_2^2\right) W_{1,2}\right] + \cdots + 2\left[\left(\nu_{n-1}^2 - 2\nu_{n-1}\nu_n + \nu_n^2\right) W_{n-1,n}\right] \\
&= \left(\nu_1^2 - 2\nu_1\nu_2 + \nu_2^2\right) W_{1,2} + \left(\nu_2^2 - 2\nu_2\nu_1 + \nu_1^2\right) W_{2,1} + \cdots + \left(\nu_{n-1}^2 - 2\nu_{n-1}\nu_n + \nu_n^2\right) W_{n-1,n} \\
&\quad + \left(\nu_n^2 - 2\nu_n\nu_{n-1} + \nu_{n-1}^2\right) W_{n,n-1} \\
2\left[\nu^T D\nu - \nu^T W\nu\right] &= \sum_{i=1}^n \sum_{j=1}^n \left(\nu_i - \nu_j\right)^2 W_{i,j} \\
\nu^T \Delta\nu &= \frac{\sum_{i=1}^n \sum_{j=1}^n \left(\nu_i - \nu_j\right)^2 W_{i,j}}{2}
\end{aligned} \tag{1}$$

Notice that $\nu^T \Delta\nu \geq 0$ because of the squared term. Thus, Δ is positive semi-definite.

2.2 Newton's Method

Derive the algorithm that solves:

$$\min_f \sum_{i=1}^n \log(1 + e^{-2y_i f_i}) + f^T \Delta f$$

Let

$$g(f) = \sum_{i=1}^n \log(1 + e^{-2y_i f_i}) + f^T \Delta f$$

We first take the gradients with respect to f:

$$\nabla_f g(f) = \sum_{i=1}^{n} \frac{-2y_i e^{-2y_i f_i}}{1 + e^{-2y_i f_i}} + 2\Delta f$$

If we define the function h such that $h(x) = \frac{e^x}{1+e^x}$ and the matrix Y such that $Y_{ij} = y_i I\{i = j\}$, we can write this as:

$$\nabla_f g(f) = -2Yh(-2yf) + 2\Delta f$$

Similarly,

$$\nabla_f^2 g(f) = \nabla_f[-2Yh(-2yf) + 2\Delta f]$$

Where:

$$\begin{aligned}\nabla_f h(-2y_i f_i) &= \frac{e^{-2y_i f_i}}{1 + e^{-2y_i f_i}}(-2y_i) - \frac{(e^{-2y_i f_i})^2}{(1 + e^{-2y_i f_i})^2}(-2y_i) \\ &= h(-2y_i f_i)(1 - h(-2y_i f_i))(-2y_i)\end{aligned}$$

So,

$$\nabla_f^2 g(f) = \sum_{i=1}^{n} -2y_i h(-2y_i f_i)(1 - h(-2y_i f_i))(-2y_i) + 2\Delta$$

We can define the matrix Z such that $Z_{ij} = h(-2y_i f_i)(1 - h(-2y_i f_i))I\{i = j\}$, then:

$$\nabla_f^2 g(f) = 4Y^T ZY + 2\Delta$$

Which gives us a Newton update step of:

$$f^{(i+1)} = f^{(i)} - \left(\nabla_f^2 g\left(f^{(i)}\right)\right)^{-1} \nabla_f g\left(f^{(i)}\right)$$

Algorithm:

Initialize $f^{(i)} = \overrightarrow{0}$
repeat
  $f^{(i+1)} = f^{(i)} - \left(\nabla_f^2 g\left(f^{(i)}\right)\right)^{-1} \nabla_f g\left(f^{(i)}\right)$
until $|f_j^{(i+1)} - f_j^{(i)}| < \epsilon \quad \forall \quad j \in 1 \ldots n$

2.3 Cora Text Data

```
> W <- as.matrix(read.table("cite.txt", header=TRUE))
> n <- nrow(W)
> D <- diag(as.vector(W %*% rep(1, n)))
> Del <- D - W
> y <- -1*(2*(as.numeric(read.table("class.txt", header=TRUE)[,1])-1)-1)
> Y <- diag(y)
> h <- function(x) exp(x)/(1 + exp(x))
> g <- function(f) log(1+exp(-2*y*f))
> g.grad1 <- function(f) -2*Y %*% h(-2*y*f) + 2*Del%*%f
> g.grad2 <- function(f) 4*t(Y) %*% diag(h(-2*y*f)*(1 - h(-2*y*f))) %*% Y + 2*Del
> f <- rep(0, n)
> eps <- 0.05
> for(i in 1:100){
+     f.i <- as.vector(f - solve(g.grad2(f)) %*% g.grad1(f))
+     if(isTRUE(all.equal(f.i, f, tolerance=eps))) break
+     f <- f.i
+ }
> y.hat <- sign(f)
> cat("Accuracy Rate: ", 100*round(table(y.hat*y)[2]/n,4), "%\n", sep="")
```

```
Accuracy Rate: 69.28%
```