# STAT 672: Homework 2

Tom Wallace

February 24, 2018

# Problem 1

## A

For a supervised learning problem, risk is defined as the expected value of the loss function:

$$R(f) = \mathbf{E}_{X,Y\sim P}[L(Y, f(x))]$$

Bayes risk is the risk present when using the Bayes classifier $f^*(x)$:

$$f^*(x) = \arg\max_Y P(Y = y|X = x)$$

Typically, this classifier is not practical because we do not know the conditional distribution of $Y$ given $X$, but in this problem it is given. Our resultant Bayes classifier is:

$$f^*(x) = \begin{cases} 0 & x \in [0.2, 0.8] \\ 1 & x \in \{(0, 0.2) \cup (0.8, 1)\} \end{cases}$$

Suppose that we use a typical 0-1 loss function.

$$L(y, f(x)) = \begin{cases} 0 & y = \text{sign}(f(x)) \\ 1 & \text{otherwise} \end{cases}$$

Risk in our problem is equal to:

$$P(\text{sign}(Y) \neq \text{sign}(f(x)))$$

Using the law of total probability, this is equal to:

$$\begin{aligned} &P(Y = 0|X \in \{(0, 0.2) \cup (0.8, 1)\})P(X \in \{(0, 0.2) \cup (0.8, 1)\}) \\ &+ P(Y = 1|X \in [0.2, 0.8])P(X \in [0.2, 0.8]) \end{aligned} \tag{1}$$

$$(0.2 \times 0.1) + (0.2 \times 0.1) + (0.2 \times 0.6) = 0.16$$

## B

First, consider $d = 0$. $f(x)$ must take the form of $f(x) = C$, with $C$ being some constant. Because $Y \in \{-1, 1\}$, our candidate functions are $f(x) = 1$ (i.e. always predict that $Y = 1$, regardless of $x$) and $f(x) = -1$ (i.e. always predict that $Y = -1$, regardless of $x$). We calculate the risk for each:

$$\begin{aligned} R(f(x)) =& P(Y \neq 1|X \in (0, 0.2))P(X \in (0, 0.2)) \\ &+ P(Y \neq 1|X \in [0.2, 0.8])P(X \in [0.2, 0.8]) \\ &+ P(Y \neq 1|X \in (0.8, 1))P(X \in (0.8, 1)) \end{aligned}$$

$$= (0.1)(0.2) + (0.8)(0.6) + (0.1)(0.2) = 0.52$$

By similar logic, $R(f(x) = -1) = 0.48$. This is smaller than 0.52 and so is the best we can do for $d = 0$. Excess risk for $d = 0$ is:

$$\min_{f\in\mathcal{F}_{d=0}} R(f) - R(f^*) = 0.48 - 0.16 = 0.32$$

Lastly, consider $d = 1$. Our candidate functions are of the form:

$$f(x) = \alpha x + C$$

This equation defines a straight line and so we are limited to a "threshold" approach: for example, always predict that $Y = 1$ if $x$ is greater than some value. Because the distribution of $Y|X$ is a simple step function,

we have some obvious candidate functions. Consider $f(x) = -x+0.2$, i.e. predict that $Y = 1$ for $X \in (0, 0.2)$ and $Y = -1$ for $X \in (0.2, 0.8)$:

$$\begin{aligned} R(f(x)) =& P(Y \neq 1|X \in (0, 0.2))P(X \in (0, 0.2)) \\ &+ P(Y \neq -1|X \in (0.2, 0.8))P(X \in (0.2, 0.8)) \\ &+ P(Y \neq -1|X \in (0.8, 1.0))P(X \in (0.8, 1.0)) \end{aligned}$$

$$= (0.1)(0.2) + (0.6)(0.2) + (0.9)(0.2) = 0.32$$

By symmetry the risk of this function is the same as other similar candidates that set the threshold at a discontinuity in the step function: we cannot do better. Excess risk for $d = 1$ thus is:

$$\min_{f \in \mathcal{F}_{d=1}} R(f) - R(f^*) = 0.32 - 0.16 = 0.16$$

Lastly, consider $d = 2$. Our candidate functions are of the form:

$$f(x) = \alpha_1 x^2 + \alpha_2 x + C$$

A good candidate is $f(x) = (x - 0.2)(x - 0.8)$. This function will return a positive number (i.e., predict that $Y = 1$) if $x \in (0, 0.2)$, and will return a negative number (i.e., predict that $Y = -1$) if $x \in (0.2, 0.8)$. Risk is thus equal to:

$$\begin{aligned} R(f(x)) =& P(Y \neq 1|X \in (0, 0.2))P(X \in (0, 0.2)) \\ &+ P(Y \neq -1|X \in (0.2, 0.8))P(X \in (0.2, 0.8)) \\ &+ P(Y \neq 1|X \in (0.8, 1.0))P(X \in (0.8, 1.0)) \end{aligned}$$

$$(0.2)(0.1) + (0.2)(0.1) + (0.2)(0.6) = 0.16$$

And excess risk is:

$$\min_{f \in \mathcal{F}_{d=2}} R(f) - R(f^*) = 0.16 - 0.16 = 0$$

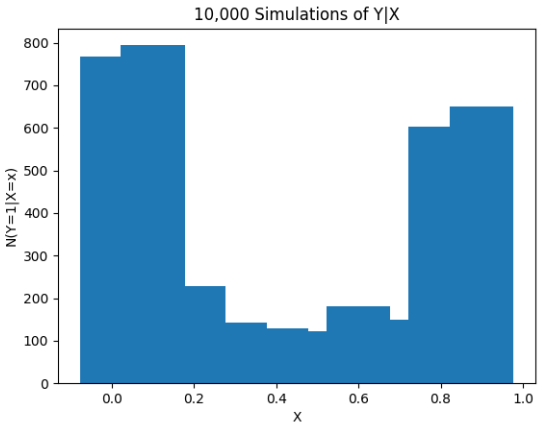
Further increasing dimensionality ($d = 3, 4 \ldots$) will allow us to do as well (i.e. achieve zero excess risk) but no better, since Bayes risk is the theoretical optimum. We have proved the following two statements and so can conclude that excess risk is zero if and only if $d \geq 2$:

- $\min_{f \in \mathcal{F}_{d \geq 2}} R(f) - R(f^*) = 0$ (if)
- $\min_{f \in \mathcal{F}_{d < 2}} R(f) - R(f^*) > 0$ (only if)

## C

See `hw2.py`. I empirically tested the accuracy of my $X$ and $Y|X$ generating functions by generating 10,000 observations with them and plotting the results, as seen in Figure 1. Although this is just a crude 10-bin histogram and so does not perfectly match the theoretical distribution, it appears broadly correct; for example, it has a "barbell" shape.

Figure 1: $X$ and $Y|X$




## D

See `hw2.py`. I trained three separate logistic regression models for $d = 1$, $d = 2$, and $d = 5$ on randomly generated $X$ and $Y$ ($n = 100$), which were generated using the specified PDFs (random uniform for $X$ and the given conditional density for $Y|X$). Table 1 shows each model's predictions for a fixed set of $X$.

Table 1: Model predictions of $Y|X$

| **X** | **d=1** | **d=2** | **d=5** |
|---|---|---|---|
| 0 | 1 | 1 | 1 |
| 0.25 | 0 | 1 | 0 |
| 0.50 | 0 | 0 | 0 |
| 0.75 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |

## E

See `hw2.py`. My results are shown in Figure 2.

As $n$ grows larger, empirical error increases and generalization error decreases. This phenomenon has an intuitive explanation.

When $n$ is small, the model has only a limited amount of information to learn on. Its classification rules overfit to the few random observations in $D$ rather than learning the true general relationship between $X$ and $Y$. Because $n$ is small, basic probability means that $D'$ likely will contain behavior not present in $D$. The model has no idea how to predict these. The model hence will do a good job of predicting $D$ (low empirical error) but a poor job of predicting $D'$ (high generalization error).

When $n$ is large, $D$ contains enough information for the model to learn the true relationship between $X$ and $Y$. It no longer will overfit to $D$-idiosyncratic behavior and so will attain higher empirical error than was present with small $N$. But, since the model has learned the true relationship between $X$ and $Y$, it also will do a better job of predicting $D'$ (lower generalization error).

Figure 2: Advanced Simulation