Conditional Probabilities, Conditional Expectations, and Loss Function

Machine Learning - Sections 2.2.1 & 2.2.2

Marc Haddad

Published: 5 January, 2020

Updated: 20 January, 2020

"The most common reason for not being able to build perfect [predictive] algorithms is that it is impossible" -Dr. Rafael Irizarry

Conditional Probabilities

Observations with the same observed values for predictors may not be the same (e.g. female with height 66 in., and male with height 66 in.; same height, different category).

However, we can assume that the observations have the same probability of being one category or another (e.g. 40% chance of having 66 in. female, 60% chance of having 66 in. male).

Mathematically represented as: $(X_1 = x_1, X_2 = x_2, ..., X_p = x_p)$ for **observed values** $(x_1, ..., x_p)$ of **covariates** $(X_1, ..., X_p)$.

We denote the conditional probabilities of each class k:

$Pr(Y = k \mid X_1 = x_1,\ X_2 = x_2, ..., X_p = x_p)$
for $k = (1, ..., K)$.

We use bold letters to write out all the predictors like this:

$\mathbf{X} = (X_1, ..., X_p)$ and
$\mathbf{x} = (x_1, ..., x_p)$.

The conditional probability of being in class k is:

$p_k(\mathbf{x}) = Pr\left(Y = k \mid \mathbf{X} = \mathbf{x}\right)$ for $k = (1, ..., K)$.

For any set of predictors $\mathbf{X}$ we will predict the class k with the largest probability among $p_1(x),\ p_2(x), ...,\ p_K(x)$ Which can be written as:

$\hat{Y} = \max_k p_k(\mathbf{x})$

However, we can't compute the above equation because we don't know the p_k of $\mathbf{x}$'s. This exemplifies the main challenge of Machine Learning: Estimating these conditional probabilities.

The better our algorithm estimates $\hat{p_k}(\mathbf{x})$,
the better our predictor $\hat{Y} = \max_k \hat{p_k}(\mathbf{x})$

The quality of our prediction will depend on two things:

1. How close the maximum probability $\max_k p_k(\mathbf{x})$ is to 1
2. How close our estimate of the probabilities $\hat{p_k}(\mathbf{x})$ are to the actual probabilities $p_k(\mathbf{x})$

Because item 1 is determined by the nature of each problem, our best option is to use item 2 to best estimate conditional probabilities.

Though it is our approach here, we must keep in mind that maximizing probability is not always optimal in practice. Our approach depends on context. **Sensitivity** and **Specificity** may differ in importance in different contexts. But having a good estimate of conditional probabilities is more often than not sufficient when building an optimal prediction model due to the fact that we can control both sensitivity and specificity.

Conditional Expectations and Loss Function

Conditional Expectations

For binary data, we can think of the conditional probability of ($Y = 1$ when $\mathbf{X} = \mathbf{x}$) as the proportion of 1's in the section of the population represented by $\mathbf{X} = \mathbf{x}$. Because the **Conditional Expectation** can be defined as the `mean` of values $(Y_1, ..., Y_n)$ in a population where Y's are either 0 or 1, this expectation is *equivalent* to the **Conditional Probability** of randomly picking a 1 among Y's (i.e. the proportion of 1's to 0's). Therefore, we mostly use conditional expectation to also represent conditional probability.

As mentioned in previous lesssons, the "correctness" of our models for *binary* outcomes can be quantified with **sensitivity**, **specificity**, **accuracy**, and the $\mathbf{F_1}$ **score**. However, these are not useful for *continous* outcomes. Generally, the best approach for validating the correctness of our continous outcomes is by using the **Loss Function**.

Loss Function

The most commonly used loss function is the **Squared Loss Function** (SLF).
Given that $\hat{Y}$ is our predictor and Y is our actual outcome, the SLF is calculated as:

$$SLF = (\hat{Y} - Y)^2$$

However, the fact we have multiple observations (N) in our test set necessitates the averaging of our multiple SLF's. This `mean` is formally known as the **Mean Squared Error** (MSE):

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (\hat{Y}_i - Y_i)^2$$

The MSE of binary outcomes is equivalent to **accuracy**, since $(\hat{Y} - Y)^2$ is either 0^* if our prediction is correct or 1^* otherwise. [*Note: Dr. Irizarry accidentally flips these values in the explanation video, I've corrected them accordingly.]

Our goal is to develop an algorithm that minimizes loss (i.e. as close as possible to 0). We must also keep in mind that the MSE itself is a random variable, due to the fact that our data (usually) comes from a random sample. This means not only do we want the the *MSE* but also the *average* of *multiple MSE*'s from *multiple samples*. We, thus, want to minimize what is referred to as the **Expectation of the Squared Error**:

$$= E\left\{\frac{1}{N}\sum_{i=1}^{N}(\hat{Y}_i - Y_i)^2\right\}$$

This is a theoretical concept because we normally only have one dataset with which to work with.

From the functions above, we see that expected value minimizes the expected square loss. Of all possible $\widehat{Y}$'s, the **conditional expectation** of Y given $\mathbf{X} = \mathbf{x}$ minimizes the expected loss (AKA the **Expectation of the Squared Error**) also given $\mathbf{X} = \mathbf{x}$. In other terms:

$$\hat{Y} = \mathrm{E}\left(Y \mid \mathbf{X} = \mathbf{x}\right) \quad \text{Minimizes} \quad \mathrm{E}\left\{(\hat{Y} - Y)^2 \mid \mathbf{X} = \mathbf{x}\right\}$$

With the above property in mind, we can say that the main task of Machine Learning is to use data to estimate conditional probabilities $f(\mathbf{x}) \equiv \mathrm{E}\left(Y \mid \mathbf{X} = \mathbf{x}\right)$, for any set of features $\mathbf{x} \equiv (x_1, ..., x_p)$.