Conditional probabilities

In machine learning applications, we rarely can predict outcomes perfectly.

Note, that spam detectors often miss emails that are clearly spam, Siri often misunderstands the words we're saying,

and your bank often thinks your card was stolen when it was not.

Instead, you are on vacation.

The most common reason for not being able to build perfect algorithms is that it is impossible.

To see this, note that most data sets will include groups of observations

with the same exact observed values for all predictors, and thus, resulting in the same prediction.

But they have different outcomes, making it impossible to make the predictions right for all these observations.

We saw a simple example of this in the previous videos for any given height, x, you will have both males and females

that are x inches tall.

So you can't predict them all right.

However, none of this means that we can't build useful algorithms that

are much better than guessing, and in some cases, better than expert opinion.

To achieve this in an optimal way, we make use of probabilistic representations of the problem.

Observations with the same observed values for the predictors may not all be the same, but we can assume

that they all have the same probability of this class or that class.

We will write this idea out mathematically for the case of categorical data.

We use the notation X1 = little x1, all the way to Xp = little xp,

to represent the fact that we have observed values, little x1 up to little xp.

(X1 = x1,...,Xp = xp)

for observed values

x1,...,xp

for covariates

X1,...,Xp

For covariates, X1 through Xp.

This does not imply that the outcome, y, will take a specific value,

but rather, that it implies a specific probability.

Specifically, we denote the "conditional probabilities of each class, k", using this notation.

Pr(Y=k | X1=x1,...,Xp=xp), for k= 1,...,K

To avoid writing out all the predictors, we'll use the following notation.

We will use bold letters to represent all the predictors like this.

**X** = (X1,..., Xp)

and

**x** = (x1,..., xp)

We will also use the following notation for the conditional probability of being in class k.

We can write it like this.

pk(**x**)= Pr(Y=k | **X** = **x**), for k = 1,...,K

Before we continue, a word of caution.

We'll be using the notation p(x) to represent conditional probabilities as functions.

Do not confuse this with the p that we use to represent the number of predictors.

Now, let's continue.

Knowing these probabilities can guide the construction of an algorithm that makes the best prediction.

For any given set of predictors, x, we'll predict the class, k, with the largest probability

among p1x, p2x, all the way up to p capital Kx.

p1(x), p2(x),... pk(x)

In mathematical notation, we can write it like this.

Y^= max pk(x)

k

But it's not this simple, because we don't know the pk of xs.

In fact, estimating these conditional probabilities can be thought of as the main challenge of machine learning.

The better our algorithm estimates p hat of kx, the better our predictor will be.

So, how good will our prediction be will depend on two things-- how close the maximum probability is to 1,

and how close our estimate of the probabilities are to the actual probabilities.

how close max pk(**x**) is to 1

k

how close our estimate

P^k(**X**) is to pk(**X**)

We can't do anything about the first restriction, as it is determined by the nature of the problem.

So our energy goes into finding ways to best estimate condition of probabilities.

The first restriction does imply that we have limits as to how well even the best possible algorithm can perform.

You should get used to the idea that while in some challenges

we will be able to achieve almost perfect accuracy-- digit readers, for example--

and others, our success is restricted by the randomness of the process-- movie recommendations, for example.

And before we continue, we note that defining our prediction

by maximizing the probability is not always optimal in practice and depends on the context.

As previously discussed, sensitivity and specificity may differ in importance in different contexts.

But even in these cases, having a good estimate of the conditional probabilities will suffice

for us to build an optimal prediction model, since we can control specificity and sensitivity however we wish.

For example, we can simply change the cutoff used to predict one class versus another.

In the plane example we gave previously, we may ground the plane anytime the probability of my function

is higher than 1 in 1,000, as opposed to the default 1/2 used when error types are equally undesired.