06.02 - Machine Learning System Design

1 Prioritizing what to work on: Spam classification example

If we want to build a spam filter, we first need to know what is the x and the y in our problem.

y is easy: 1 if spam, 0 if ham.

One way of choosing our x features would be to find 100 words indicative of spam/ham. Then, our feature vector would be full of boolean features each indicating whether the word appeared in the email or not.

1.1 How to spend your time to make it have low error?

- Collect data. Eg “honeypot” project.
- Sophisticated features based on email header.
- Sophisticated features based on email body. Eg should “discount” and “discounts” be treated as the same word.
- Develop algorithm to detect misspellings (med1cine, w4tches).

2 Error Analysis

2.1 Recommended approach

- Start with a simple algorithm that you can implement quickly (avoid premature optimization).
- Plot learning curves to decide if more data, more features are likely to help.
- Error analysis: Manually examnie the examples that your algorithm made errors on. See if you spot any trend in the type of examples it is making errors on.

In the case of the emails, if we have a certain amount of missclassified emails, we can check the type of them (eg most common are pharma, replica, steal passwords, ...) and what features you thjink would have helped the algorithm to classify them correctly.

Manually doing this can serve the purpose of finding the thing where spending time is more worth it.

2.2 The importance of numerical evaluation

It’s important to have a reference number (such as accuracy) in order to know whether doing a change is an improvement or not.

Eg, should discount/discounts/discounted/discounting be treated as the same word? Check score doing it / not doing it and decide based on that.

3 Error metrics for skewed classes

3.1 Cancer classification example

- Logistic regression model.
- 1% Error on test set.

Problem: 0.5% of patients have cancer. An algorithm that simply says that y = 0 (no cancer) has a 0.5% of error!!

Skewed classes = when we have a lot more data from a class than the other one.

3.2 Different error metrics

In this cases, we need to use different error metrics (confusion matrix):

- Precision: Of all patients where we predicted y = 1, what fraction actually has cancer? Equation 1.
- Recall: Of all patients that actually have cancer, waht fraction did we correctly detect as having cancer? Equation 2.

$$\text{precision} = \frac{\text{True positives}}{\text{\# predicted positives}} \tag{1}$$

$$\text{recall} = \frac{\text{True positives}}{\text{\# actual positives}} \tag{2}$$

So, in the case of the model that simply says y = 0, we could use recall. The result would be a recall of zero, which we will then try to improve.

y = 1 in presence of the most **rare class**.

4 Trading Off Precision and Recall

In the example of the cancer algorithm, we want to be very confident as it's very critical to fail.

Suppose we want to predict y = 1 only if very confident. If we're using logistic regression, we can simply cut the threshold in which we decide whether someone has cancer at a higher %.

Doing that, we get higher precision and lower recall.

On the other hand, if we want to avoid mising cases of cancer (avoid false negatives), we can just cut the threshold lower so more people are classified as having cancer.

We will have lower precision and higher recall.

We can see this in figure 1.

4.1 F score

A way to compare precision and recall, as given in equation 3.

$$\text{f score} = 2\frac{PR}{P+R} \tag{3}$$

5 Data For Machine Learning

> It's not the best algorithm that wins, but the one with more data.

We must make sure the features we have have sufficient info to predict y accurately, though. A good way to tell that is thinking about whether a human expert could predict y with the data we have.

5.1 Large data rationale

Pick an algorithm with many parameters so that it can learn difficult functions.

Also use a large training set (unlikely to overfit).

Doing this we will have a small train error and the train error will be more or less equal to the test error. That means that the train error will be small. That is assuming we have a loooot of data.

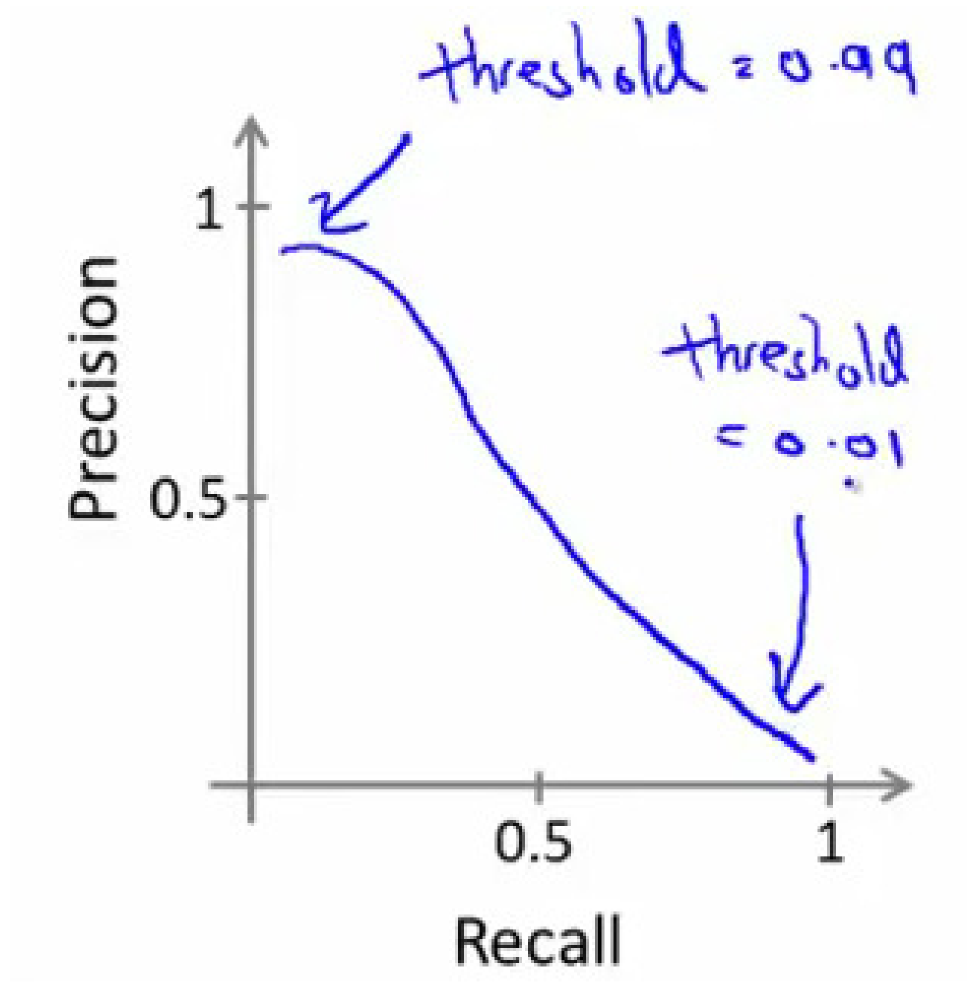

Figure 1: Precision and recall depending on the threshold we cut at.