Machine Learning by Andrew Ng
pdf by CarolusRex

Machine learning system design

Prioritizing what to work on: Spam classification example

After we build a spam classifier, how to make it have low error:

1. Collect lots of data
   E.g. “honeypot” project.
2. Develop sophisticated features based on email routing information (from email header).
3. Develop sophisticated features for message body,
   e.g. should “discount” and “discounts” be treated as the same word? How about “deal” and “Dealer”? Features about punctuation?
4. Develop sophisticated algorithm to detect misspellings (e.g. m0rtgage, med1cine, w4tches.)

Error analysis

1. Start with a simple algorithm that you can implement quickly. Implement it and test it on your cross--validation data.
2. Plot learning curves to decide if more data, more features, etc. are likely to help.
3. Error analysis: Manually examine the examples (in cross validation set) that your algorithm made errors on. See if you spot any systematic trend in what type of examples it is making errors on.

Error metrics for skewed classes

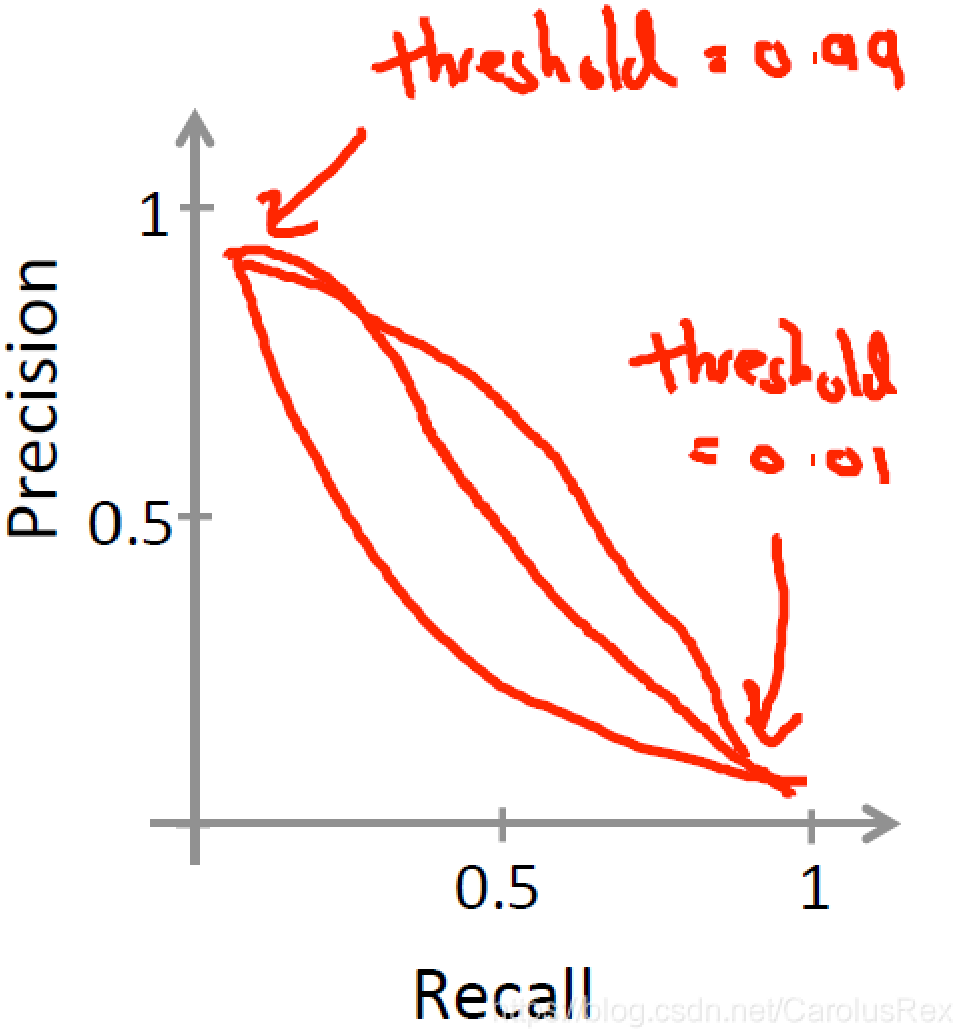
	Actual Class	**1**	**0**

	Actual Class	**1**	**0**
Predicted Class			
1		True Positive	False Positive
0		False Negative	True Negative

Precision: Of all patients where we predicted $y == 1$, what fraction actually has cancer?)
$\frac{True\ Positive}{\#predicted\ Positive} = \frac{True\ Positive}{True\ Positive + False\ Positive}$

Recall: (Of all patients that actually have cancer, what fraction did we correctly detect as having cancer?)
$\frac{True\ Positive}{\#actual Positive} = \frac{True\ Positive}{True\ Positive + False\ Negative}$

Trading off precision and recall



Suppose we want to predice $y == 1$(cancer) only if very confident. $\rightarrow$ High precision and low recall.
Suppose we want to avoid missing too many cases of cancer. $\rightarrow$ High recall and low precision.

skewd classes

$F_1 = 2\frac{PR}{P+R}$
In general, large data is unlikely to overfit.