May 21, 2021

x = features of email
y = spam (1) or not spam (0)

Features x: Choose 100 words indicative of spam/not spam.

e.g. $x = \begin{bmatrix} 0 \\ | \\ | \\ 0 \\ \vdots \end{bmatrix}$ Charles, buy, deal, plans

✗ In practice, you would take most frequently occurring words (10k - 50k) in training set.

How to Improve Accuracy?

→ Collect lots of data (this doesn't always work)
→ Develop sophisticated features (e.g. Incorporating email header data into spam classifier)
→ Develop algos to process input in different ways (e.g. Recognizing misspellings in spam.)
✗ Difficult to tell which will be most helpful

Error Analysis:

A recommended approach to solving ml problems is:

- start w/ a simple algo, implement it quickly, test on validation set
- plot learning curves to see if more data, features will help
- manually examine errors on examples in validation set to spot a trend where most errors were made

e.g. $m_{cv} = 500$

Algo misclassifies 100 emails.

→ Manually examine the 100 errors, categorize them based on:

i) Type of email. (e.g. pharma, fake, steal passwords)

ii) What features you think would have helped the algo classify them correctly. (e.g. misspellings, unusual email routing)

Numerical Evaluation:

Sometimes error analysis may not be helpful for deciding if a technique is likely to improve performance. Sometimes you just need to test it directly.

e.g. Stemming software

→ If validation error drops 2% w/ stemming, then it should be added to model.