

Evaluating a Learning Algorithm

- Video: Deciding What to Try
 Next
 5 min
- Video: Evaluating a Hypothesis 7 min
- Reading: Evaluating a Hypothesis 4 min
- Video: Model Selection and Train/Validation/Test Sets
 12 min
- Reading: Model Selection and Train/Validation/Test Sets
 3 min

Bias vs. Variance

- Video: Diagnosing Bias vs.
 Variance
 7 min
- Reading: Diagnosing Bias vs. Variance
 3 min
- Video: Regularization and Bias/Variance
 11 min
- Reading: Regularization and Bias/Variance
 3 min
- Video: Learning Curves
 11 min
- Reading: Learning Curves
 3 min
- Video: Deciding What to Do Next Revisited 6 min
- Reading: Deciding What to do Next Revisited
 3 min

Review

- Reading: Lecture Slides
 10 min
- Quiz: Advice for Applying Machine Learning
 5 questions
- Regularized Linear
 Regression and
 Bias/Variance
 3h

Building a Spam Classifier

- Video: Prioritizing What to Work On 9 min
- Reading: Prioritizing What to Work On
 3 min

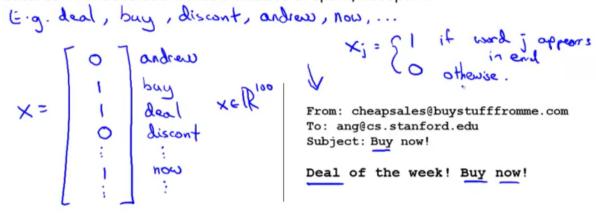
Prioritizing What to Work On

System Design Example:

Given a data set of emails, we could construct a vector for each email. Each entry in this vector represents a word. The vector normally contains 10,000 to 50,000 entries gathered by finding the most frequently used words in our data set. If a word is to be found in the email, we would assign its respective entry a 1, else if it is not found, that entry would be a 0. Once we have all our x vectors ready, we train our algorithm and finally, we could use it to classify if an email is a spam or not.

Building a spam classifier

Supervised learning. $\underline{x} = \text{features of email.} \ y = \text{spam (1) or not spam (0)}.$ Features x: Choose 100 words indicative of spam/not spam.



So how could you spend your time to improve the accuracy of this classifier?

- Collect lots of data (for example "honeypot" project but doesn't always work)
- Develop sophisticated features (for example: using email header data in spam emails)
- Develop algorithms to process your input in different ways (recognizing misspellings in spam).

It is difficult to tell which of the options will be most helpful.





