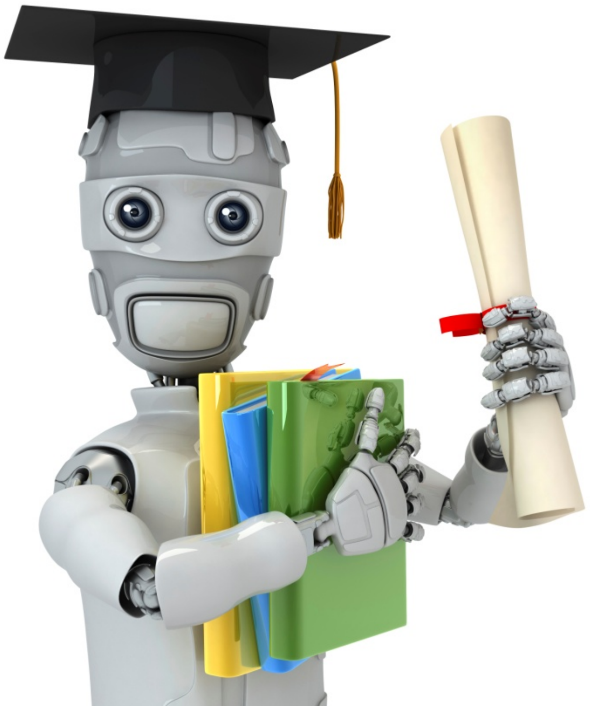


Machine Learning

Machine learning system design

Prioritizing what to work on: Spam classification example

Building a spam classifier

```
From: cheapsales@buystufffromme.com
To: ang@cs.stanford.edu
Subject: Buy now!

Deal of the week! Buy now!
Rolex w4tchs - $100
Med1cine (any kind) - $50
Also low cost M0rgages
available.
```

Spam (1)

```
From: Alfred Ng
To: ang@cs.stanford.edu
Subject: Christmas dates?

Hey Andrew,
Was talking to Mom about plans
for Xmas. When do you get off
work. Meet Dec 22?
Alf
```

Non-spam (0)

Building a spam classifier

Supervised learning. $x =$ features of email. $y =$ spam (1) or not spam (0).

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

E.g. deal, buy, discont, andrew, now, . . .

$$x = \begin{array}{c|l} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 1 \\ \vdots \end{bmatrix} & \begin{array}{l} \text{andrew} \\ \text{buy} \\ \text{deal} \\ \text{discont} \\ \vdots \\ \text{now} \\ \vdots \end{array} \end{array} \qquad x \in \mathbb{R}^{100}$$

$$x_j = \begin{cases} 1 & \text{if word } j \text{ appears in email} \\ 0 & \text{otherwise.} \end{cases}$$

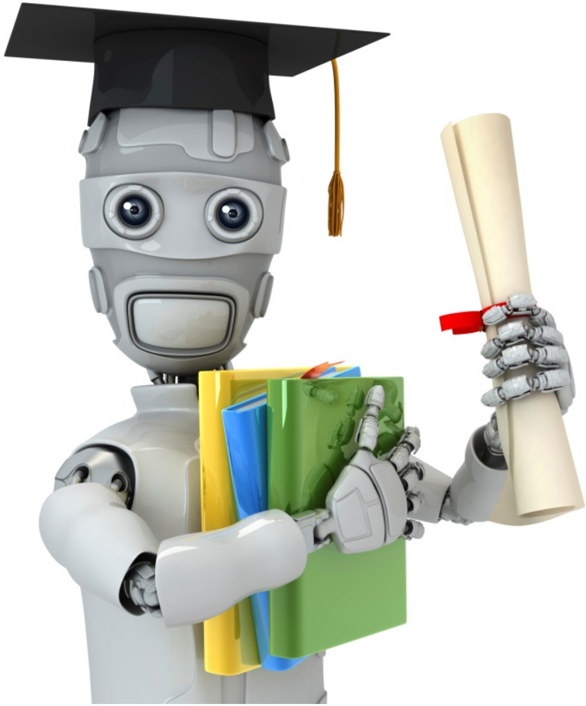
```
From: cheapsales@buystufffromme.com
To: ang@cs.stanford.edu
Subject: Buy now!

Deal of the week! Buy now!
```

Note: In practice, take most frequently occurring n words (10,000 to 50,000) in training set, rather than manually pick 100 words.

Building a spam classifier

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

- Collect lots of data
    - E.g. "honeypot" project.
- Develop sophisticated features based on email routing information (from email header).
- 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?
- Develop sophisticated algorithm to detect misspellings (e.g. m0rtgage, med1cine, w4tches.)



Machine Learning

Machine learning system design

Error analysis

Recommended approach

Good

- Start with a simple algorithm that you can implement quickly. Implement it and test it on your cross-validation data.
- Plot learning curves to decide if more data, more features, etc. are likely to help.
- 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 Analysis

$m_{CV} =$ 500 examples in cross validation set
Algorithm misclassifies 100 emails.
Manually examine the 100 errors, and categorize them based on:

- (i) What type of email it is pharma, replica, steal passwords, ...
- (ii) What cues (features) you think would have helped the algorithm classify them correctly.

Pharma: 12
Replica/fake: 4
Steal passwords: 53
Other: 31

Deliberate misspellings: 5
(m0rgage, med1cine, etc.)
Unusual email routing: 16
Unusual (spamming) punctuation: 32

Andrew Ng

The importance of numerical evaluation

Should discount/discounts/discounted/discounting be treated as the same word?

stemming software就是比較兩個單詞的前面的字母，若相同就認為它們意思一樣，但也會將下面的universe和university認為意思一樣.

Can use "stemming" software (E.g. "Porter stemmer")

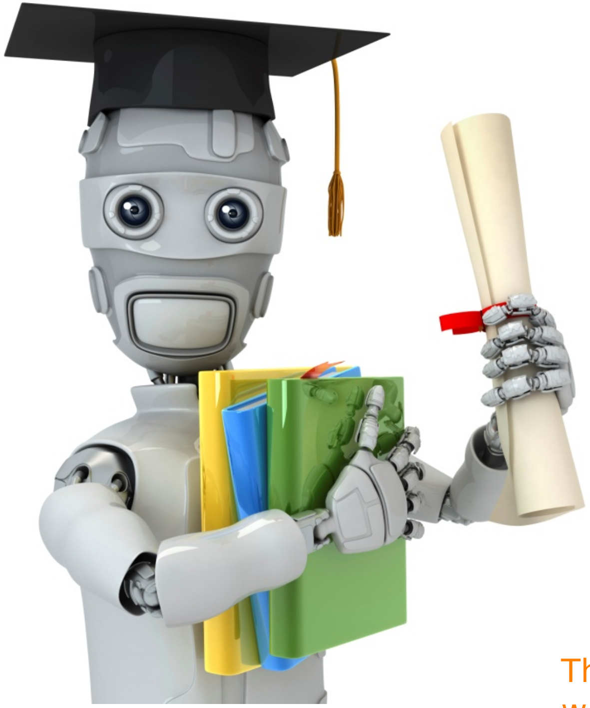
universe/university.

Error analysis may not be helpful for deciding if this is likely to improve performance. Only solution is to try it and see if it works.

Need numerical evaluation (e.g., cross validation error) of algorithm's performance with and without stemming.

Without stemming: 5% error With stemming: 3% error

Distinguish upper vs. lower case (Mom/mom): 3.2%



Machine Learning

Machine learning system design

Error metrics for skewed classes

There is one important case, where it's particularly tricky to come up with an appropriate error metric, or evaluation metric, for your learning algorithm. That case is the case of what's called skewed classes.

Cancer classification example

Train logistic regression model $h_\theta(x)$. ($y = 1$ if cancer, $y = 0$ otherwise)

Find that you got $\boxed{\text{1\% error}}$ on test set.

(99% correct diagnoses)

Only 0.50% of patients have cancer.

→ skewed classes.

```
function y = predictCancer(x)
  → y = 0; %ignore x!
return
```

→ 0.5% error

→ 99.2% accuracy (0.8% error)

→ 99.5% accuracy (0.5% error)

Precision/Recall

$y = 1$ in presence of rare class that we want to detect

Actual class

Predicted class	1	0
1	True positive	False positive
0	False negative	True negative

$y = 0$

recall $= 0$

Precision

(Of all patients where we predicted $y = 1$, what fraction actually has cancer?)

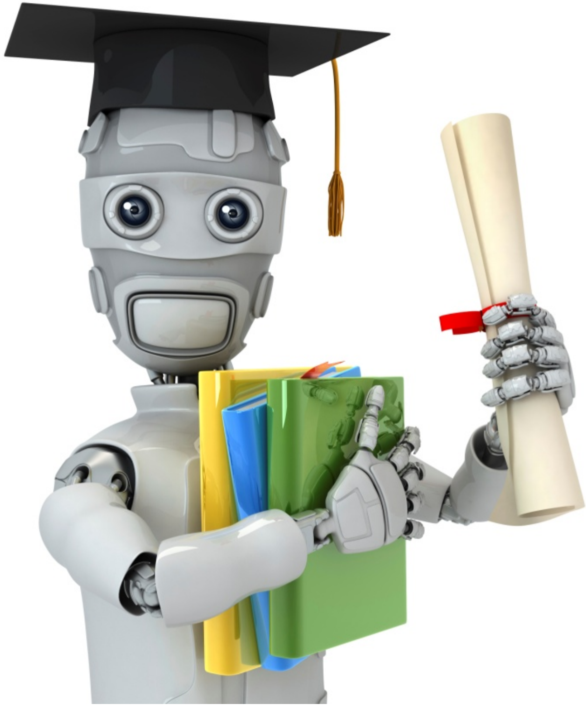
True postivie是指所有預測為pos中的有多少個是真正的pos

$$\frac{\text{True positives}}{\#\text{predicted positive}} = \frac{\text{True positive}}{\text{True pos} + \text{False pos}}$$

Recall

(Of all patients that actually have cancer, what fraction did we correctly detect as having cancer?)

$$\frac{\text{True positives}}{\#\text{actual positives}} = \frac{\text{True positives}}{\text{True pos} + \text{False neg}}$$



Machine Learning

Machine learning system design

Trading off precision and recall

Trading off precision and recall

precision = $\frac{\text{true positives}}{\text{no. of predicted positive}}$

recall = $\frac{\text{true positives}}{\text{no. of actual positive}}$

Logistic regression: $0 \leq h_\theta(x) \leq 1$

Predict 1 if $h_\theta(x) \geq \cancel{0.5}$ ~~0.7~~ ~~0.9~~ 0.3

Predict 0 if $h_\theta(x) < \cancel{0.5}$ ~~0.7~~ ~~0.9~~ 0.3

Suppose we want to predict $y = 1$ (cancer) only if very confident.

h_theta(x) > 0.7

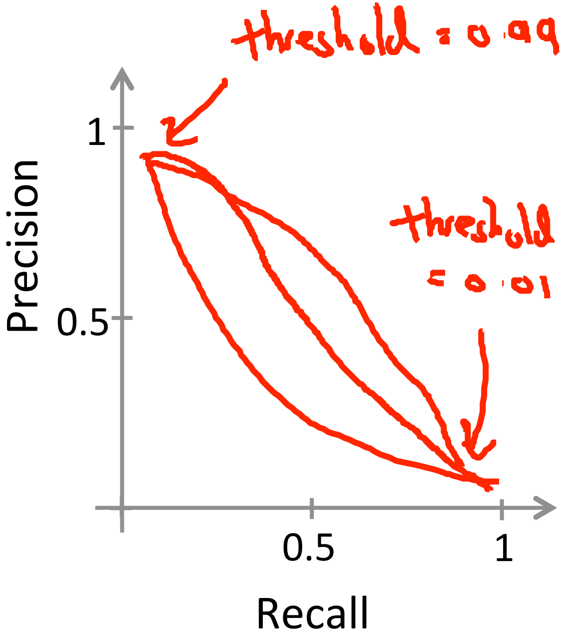
→ Higher precision, lower recall

(原字幕只有中文的)这个回归模型会有较低的召回率 因为 当我们做预测的时候 我们只给很小一部分的病人预测y=1

Suppose we want to avoid missing too many cases of cancer (avoid false negatives).

→ Higher recall, lower precision. h_theta(x) > 0.3

More generally: Predict 1 if $\boxed{h_\theta(x) \geq \text{threshold}}$



一个有非常高的临界值(即h_theta(x)的threshold) 一个有非常低的临界值 它们中的任何一个都不是一个好的模型

F_1 Score (F score)

How to compare precision/recall numbers?

	Precision(P)	Recall (R)
Algorithm 1	0.5	0.4
Algorithm 2	0.7	0.1
Algorithm 3	0.02	1.0

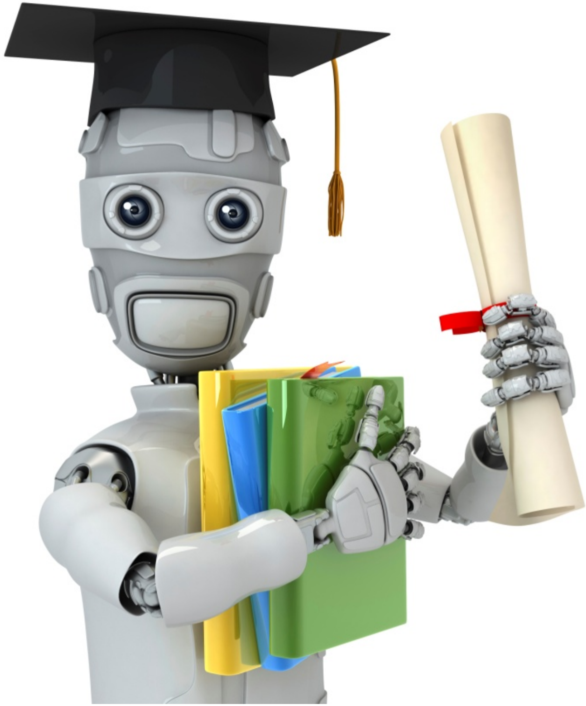
Predict y=1 all the time

Average: $\cancel{\dfrac{P+R}{2}}$

$$\boxed{F_1 \text{ Score: } \quad 2\frac{PR}{P+R}}$$

P=0 or R=0 ⇒ F-score = 0

P=1 and R=1 ⇒ F-score = 1



Machine Learning

Machine learning system design

Data for machine learning

他们当时考虑这样一个问题 如何在易混淆的词之间进行分类　并尝试将其分类 什么样的词 在一个英文句子特定的位置 才是合适的

Designing a high accuracy learning system

他们所做的就是 改变了训练数据集的大小 并尝试将这些学习算法 用于不同大小的 训练数据集中 这就是他们得到的结果

E.g. Classify between confusable words.

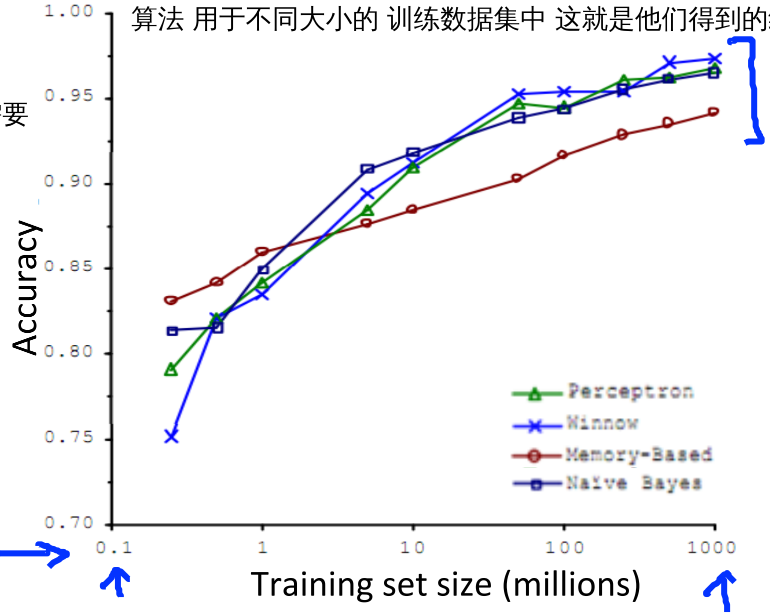
{to, two, too}, {then, than}

告诉我 中间我需要填的 词是two而不是单词 to 或 too

For breakfast I ate ______ eggs. two

Algorithms　他们用了几种不同的学习算法

- Perceptron (Logistic regression)
- Winnow
- Memory-based
- Naïve Bayes



"It's not who has the best algorithm that wins.

It's who has the most data."

那么这种说法 在什么时候是真 什么时候是假呢？

[Banko and Brill, 2001]

Large data rationale

Assume feature $x \in \mathbb{R}^{n+1}$ has sufficient information to predict y accurately.

~~to~~, ~~too~~

Example: For breakfast I ate two ______ eggs.

Counterexample: Predict housing price from only size (feet²) and no other features.

Useful test: Given the input x, can a human expert confidently predict y?

Large data rationale

Use a learning algorithm with many parameters (e.g. logistic regression/linear regression with many features; neural network with many hidden units). low bias algorithms.

$J_{train}(\theta)$ will be small.

Use a very large training set (unlikely to overfit) low variance

$J_{train}(\theta) \approx J_{test}(\theta)$

we have such a massive training set and by unlikely to overfit what that means is that the training error will hopefully be close to the test error.

$J_{test}(\theta)$ will be small

Finally putting these two together that the train set error is small and the test set error is close to the training error what this two together imply is that hopefully the test set error will also be small.

we want it not to have high bias and not to have high variance.