

The importance of numerical evaluation

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

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%

Cancer classification example

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

Find that you got 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% accy (0.8% error)
→ 99.5% accy (0.5% error)

Precision/Recall

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

Actual class	
Predicted class	1
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?)

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

Recall

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

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



09:41 / 11:35



Trading off precision and recall

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

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

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

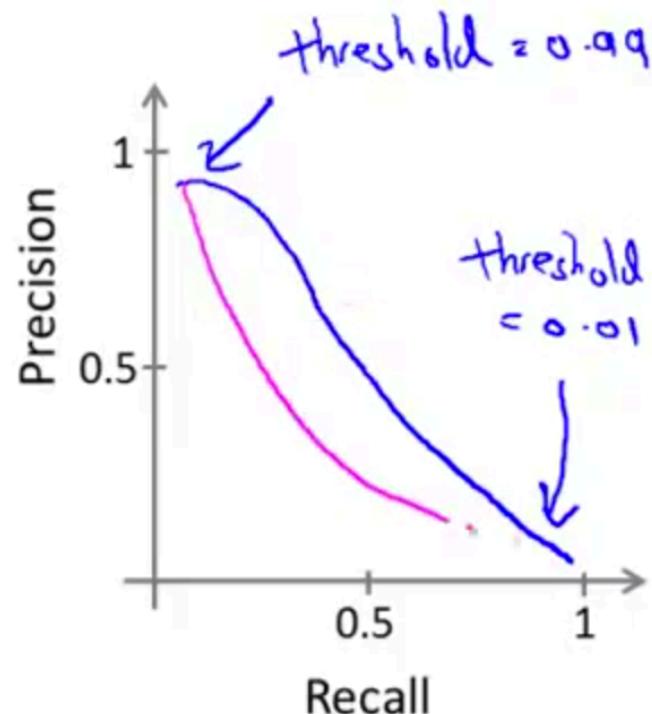
- Suppose we want to predict $y = 1$ (cancer) only if very confident.

→ Higher precision, lower recall.

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

→ Higher recall, lower precision.

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



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

F₁ Score (F score)

How to compare precision/recall numbers?

	Precision(P)	Recall (R)	Average	F ₁ Score
Algorithm 1	0.5	0.4	0.45	0.444 ←
Algorithm 2	0.7	0.1	0.4	0.175 ←
Algorithm 3	0.02	1.0	0.51	0.0392 ←

Average: ~~$\frac{P+R}{2}$~~

Predict $y=1$ all the time

F₁ Score: $2 \frac{PR}{P+R}$

Designing a high accuracy learning system

E.g. Classify between confusable words.

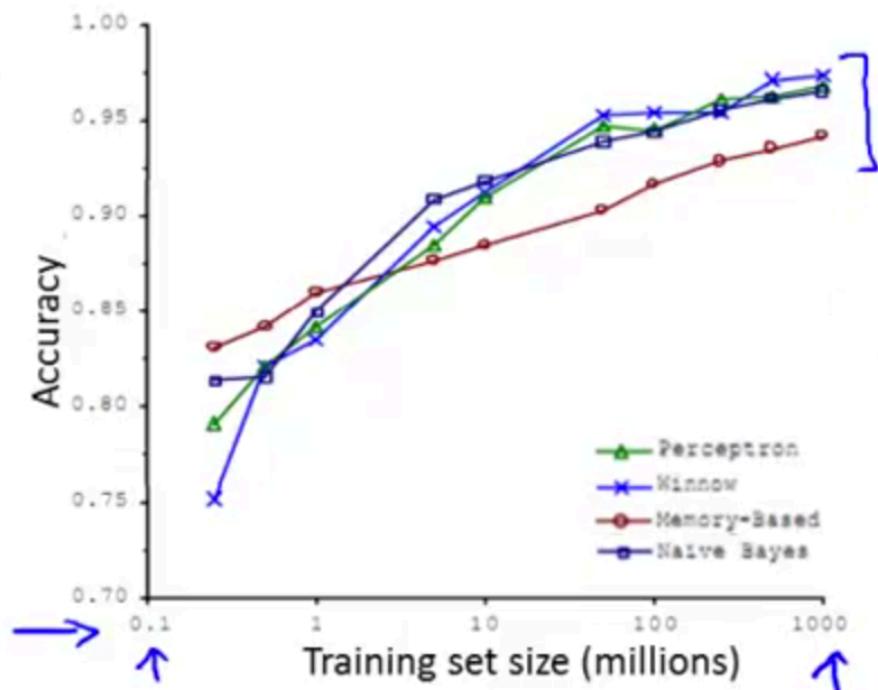
{to, two, too}

{then, than}

→ For breakfast I ate two eggs.

Algorithms

- - 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.”

[Banko and Brill, 2001]

Large data rationale

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

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 ?



07:38 / 11:09



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.

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

Use a very large training set (unlikely to overfit)

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

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

For reference:

- Accuracy = (true positives + true negatives) / (total examples)
- Precision = (true positives) / (true positives + false positives)
- Recall = (true positives) / (true positives + false negatives)
- F_1 score = $(2 * \text{precision} * \text{recall}) / (\text{precision} + \text{recall})$