Evaluating a Classification Algorithm

Evaluation

- How can we tell if our ML algorithm is doing well?
- For estimating values (like in linear regression) we can just compute statistics on the error
- How about with classifiers?
  - We can count how often we predict the correct class
  - We call this *accuracy*.
    - Let Y_i be the true class, and $\widehat{Y}_i$ be the predicted class for observation i.

$$accuracy = \frac{1}{N}\sum_{i=1}^{N}(Y_i = \widehat{Y}_i)$$

- In addition, there are two types of classification:
  - Binary classification – Just two classes
  - Multi-Class classification – More than two classes
- Each of these have their own additional evaluations of interest.

Binary Classification Error Types

- For binary classification, often we're focused on attempting to "find" one particular class.
  - We refer to this as the positive class.
  - The other data is called the negative class.
- From this we can describe four different possibilities:
  - True positive = Hit
  - True negative = Correct rejection
  - False positive = False Alarm (Type 1 error)
  - False negative = Miss (Type 2 error)

	Predicted positive	Predicted negative
Positive examples	True positives	False negatives
Negative examples	False positives	True negatives

Evaluating your Classifier

- From the four error types, we can establish some binary-classification-specific measurements:
- *Precision* – percentage of things that were classified as positive and actually were positive
  - $Precision = \frac{TP}{TP+FP}$
- *Recall* – the percentage of true positives (*sensitivity*) correctly identified
  - $Recall = \frac{TP}{TP+FN}$
- *f-measure* – The weighted harmonic mean of precision and recall
  - $F_1 = \frac{2*precision*recall}{precision+recall}$

Using Class Likelihood

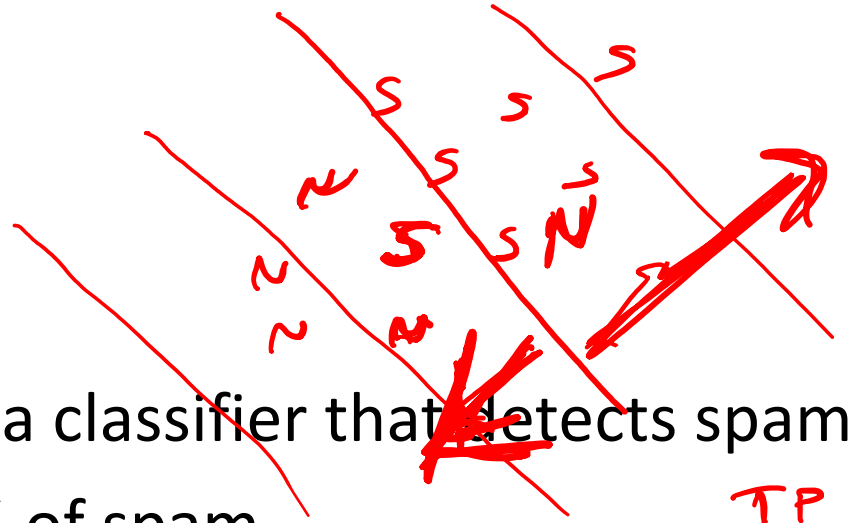
- Some classifiers don't just return what class an observation belongs to, but also return the probability of belonging to that class:

$$P(y = i|x)$$

- In these cases, we can use a *threshold* to determine what class an observation belongs to.
- For instance, for binary classification we can say:

$$\hat{y} = \begin{cases} Positive & P(y = Positive|x) > t \\ Negative & otherwise \end{cases}$$

Spam Example



- Let's imagine creating a classifier that detects spam
- It's easy to catch 100% of spam
  - Throw out ALL the mail
  - Set "threshold" to 0 (or 1)
- It's easy to make no mistakes on good mail
  - Keep ALL the mail
- Perhaps a good starting point is to choose 50% threshold
  - Anything above this is a "positive hit"
  - Anything below this is a "negative rejection"

Precision $\frac{TP}{TP+FP}$ ↓

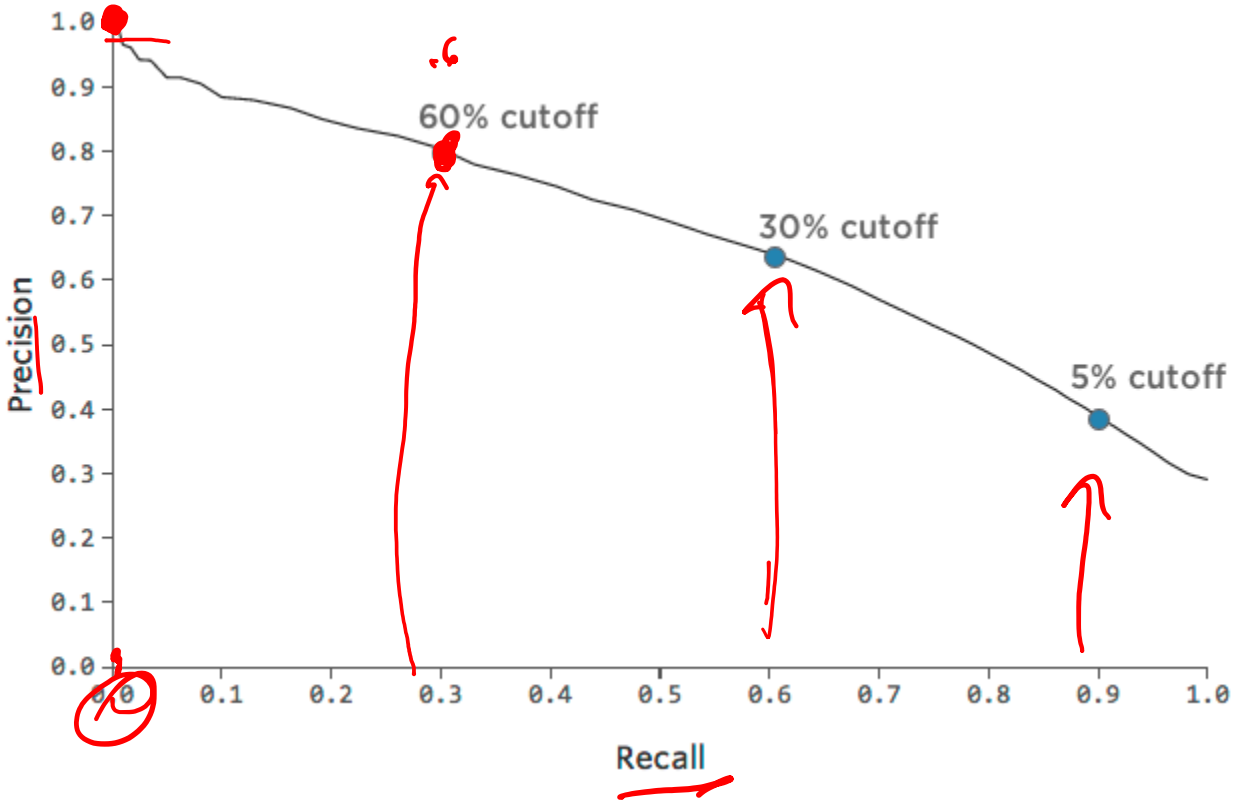
Recall $\frac{TP}{TP+FN}$ ↑

Precision/Recall Tradeoff

- We can explore the effect of this threshold on the precision and recall values.
- The plot of precision vs recall as a function of the threshold creates something called a *precision-recall* curve (PR)

Precision/Recall Curve

t [0, 0.1, 0.2 … 0.9, 1]

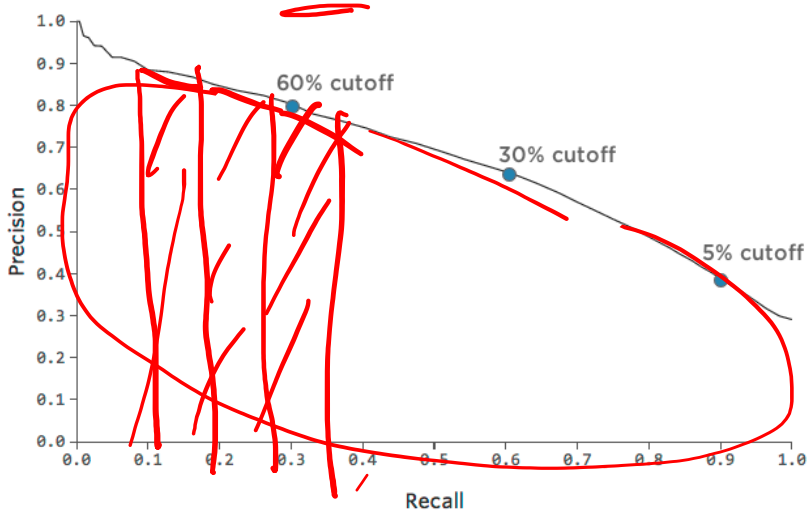


Precision/Recall Curve

- To evaluate a binary classifier, we can also compute the *area under the curve (AUC)* of a PR curve
- Given points on the curve, (R_k, P_k) we can approximate the AUC as:

$$AUC = 1 - \frac{1}{2}\sum_{k=1}^{n}(P_k + P_{k-1})(R_k - R_{k-1})$$

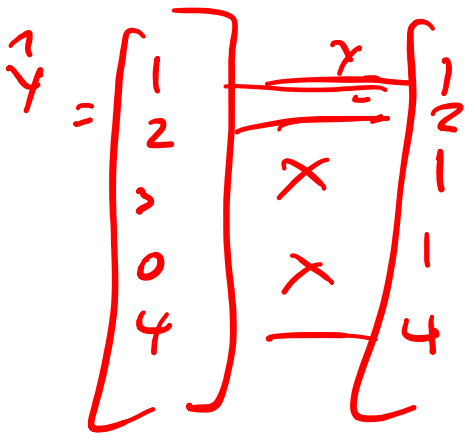
- An ideal PR curve will have an AUC of 1.0



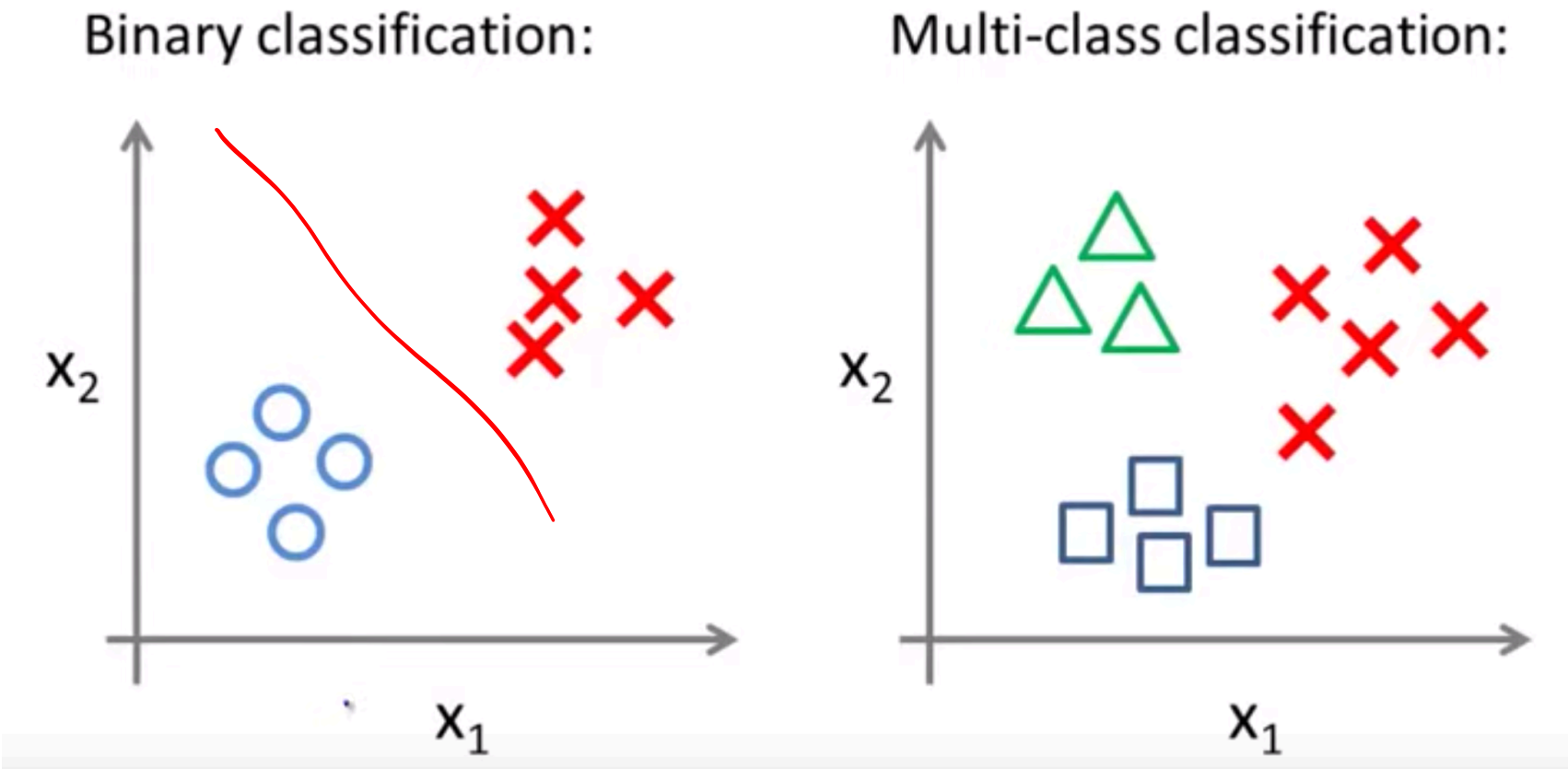
Multi-Class Evaluation

- Just like binary classification, we can evaluate the accuracy of a multi-class classifier:

$$accuracy = \frac{1}{N}\sum_{i=1}^{N}(Y_i = \hat{Y}_i)$$

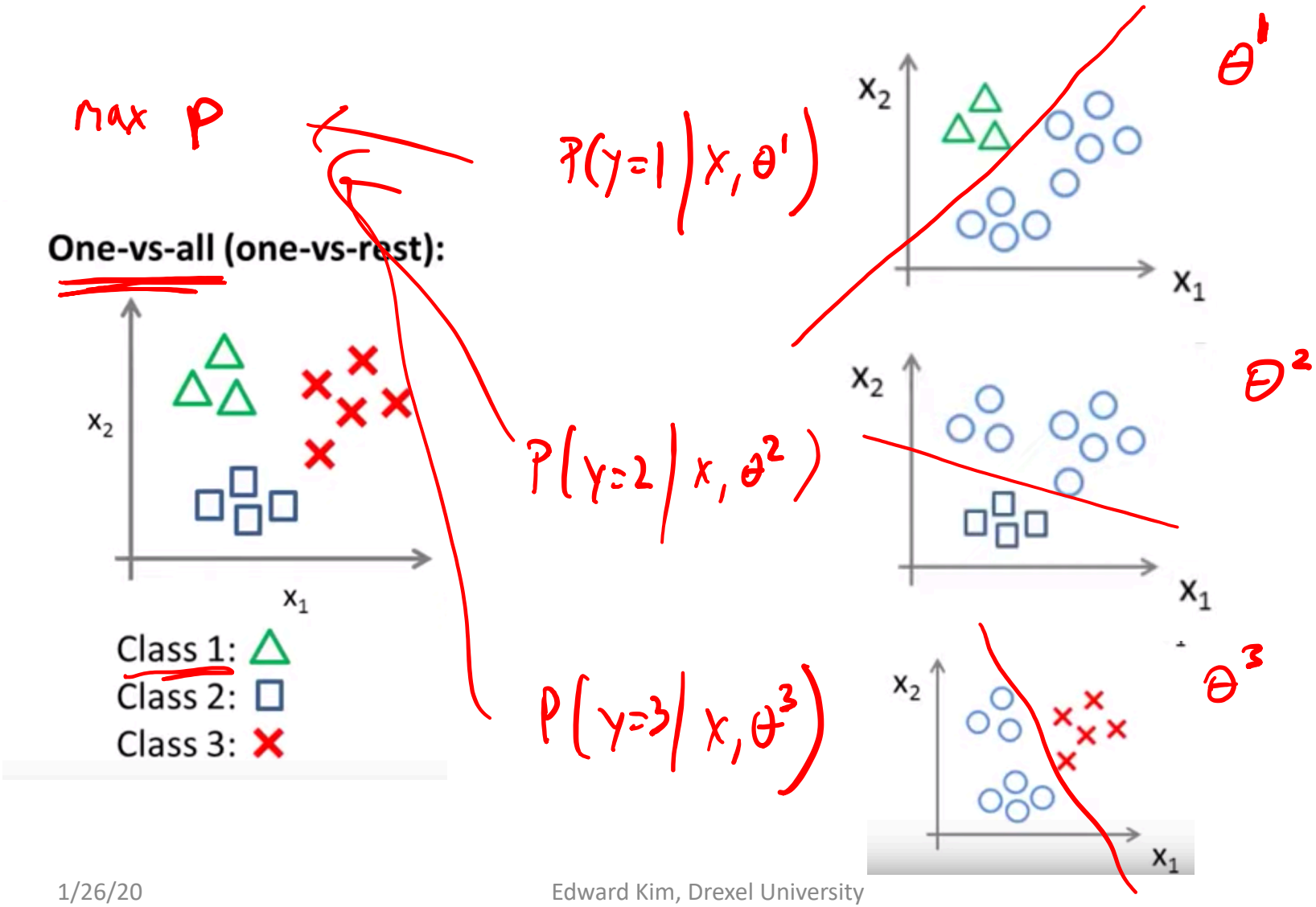


Multiclass classification



From Andrew Ng

Multiclass classification



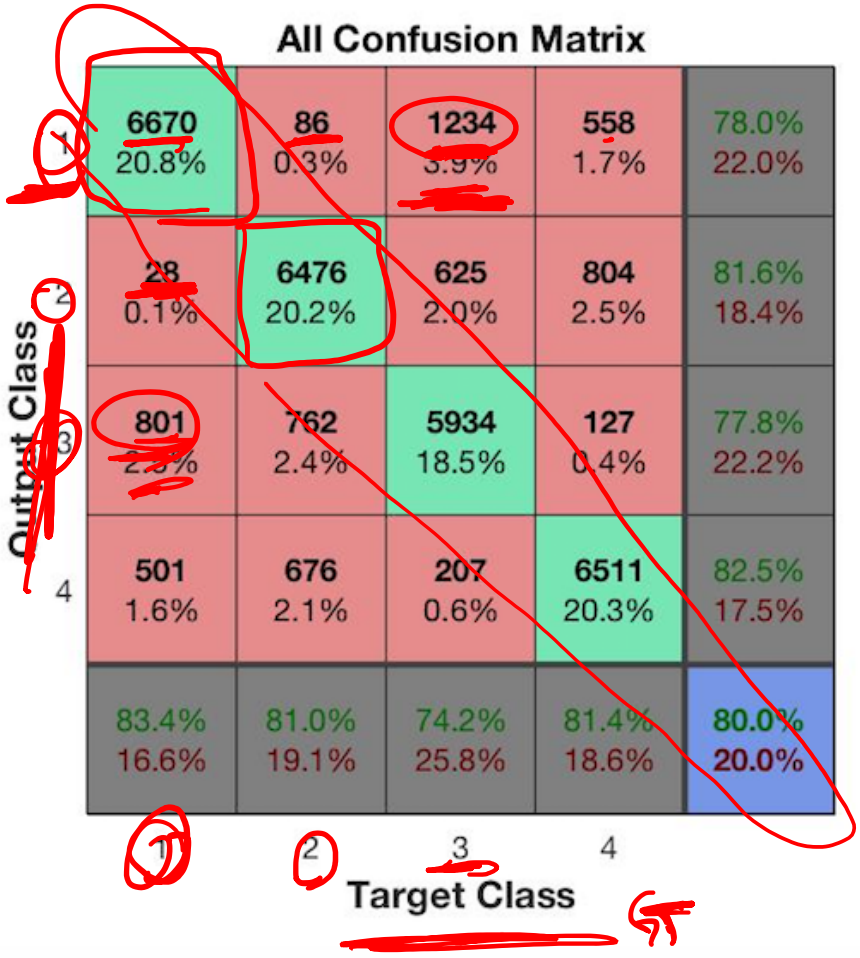
Multi-Class Evaluation

- Just like binary classification, we can evaluate the accuracy of a multi-class classifier:

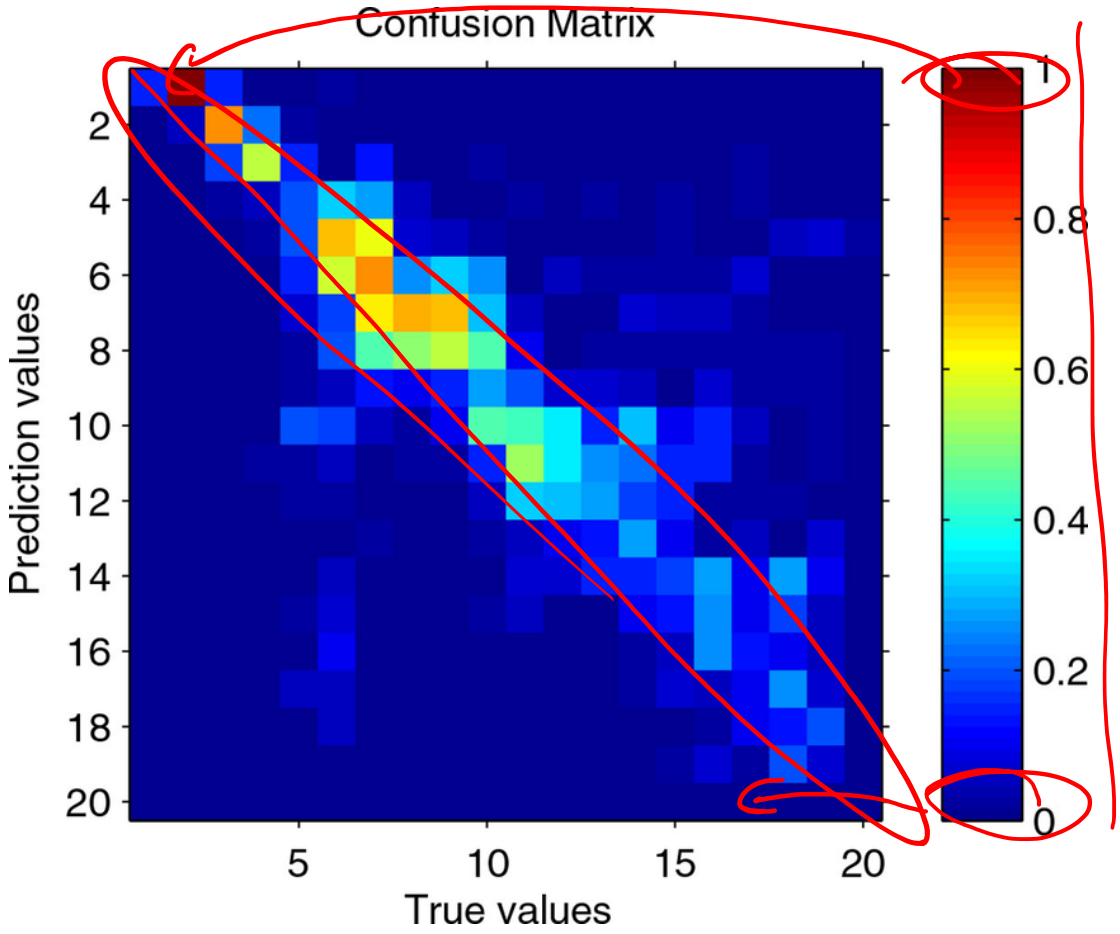
$$accuracy = \frac{1}{N}\sum_{i=1}^{N}\left(Y_i = \widehat{Y}_i\right)$$

- In addition, particular to multi-class classification, we may be interested in investigating which classes get confused with which other classes
- To observe this, we can look at a *confusion matrix*

Confusion Matrix



Confusion Matrix



Statistical Learning

Statistical Learning

- For another approach to classification we will look at the probability and statistics of our labeled data to make predictions on new data
- Review the Prob/Stats Week 0 slides!
- Hopefully these methods matches some of your intuition about data

Inference

training → θ's

testing → new unseen → inference

- Our statistical learning will start with the concept of *inference*
  - Given distribution of seen data, what can we *infer* about new data?
- Given evidence/features $x = [x_1, x_2, \dots, x_D]$ what is the *likelihood* that our object came from class i?
  - This is written as:
    $P(y = i | feature_1 = x_1, feature_2 = x_2, \dots, feature_D = x_D)$
  - We'll just abbreviate this as:

$$P(y = i | f_1 = x_1, f_2 = x_2, \dots, f_D = x_D) = P(y = i | f = x)$$

$P(y|x)$

- We call this value $P(y = i | f = x)$ that we're trying to compute, the *posterior*

$P(y = i)$ prior

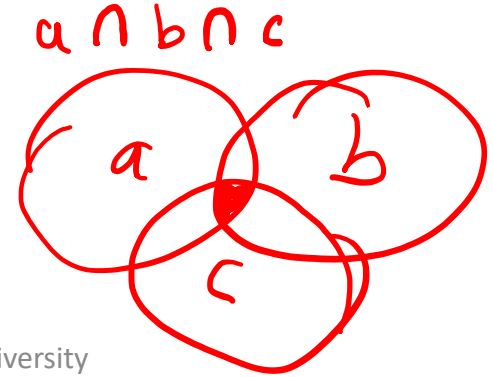
Inference

$$P(y = i | f_1 = x_1, f_2 = x_2, \ldots, f_D = x_D)$$

- Recall from probability that this is a *conditional probability:*

  "Given the first feature has value x_1, the second has x_2, etc.. what is the probability that our class was i?"

- Also recall that $P(a, b, c) = P(a \wedge b \wedge c)$ is called the *joint probability.*



Inference

- From the rules of probability

$$P(y|x) = \frac{P(y \wedge x)}{P(x)} = \frac{P(y,x)}{P(x)}$$

  - Here we call $P(x)$ the *evidence*.

- So we can solve this inference problem as:

$$P(y = i|f_1 = x_1, .., f_D = x_D) = \frac{P(y = i, f_1 = x_1, \dots, f_D = x_D)}{P(f_1 = x_1, \dots f_D = x_D)}$$

- Our final probability is defined purely in terms of the joints
  - And given enough data we be able get the joints easily directly from our data!

The Joint Distribution

$2^3 = 8$

- How to make a joint distribution:

1. Make a truth table listing all combinations of values of your variables (if there are M Boolean variables, then the table will have 2^M rows)
2. Count how many times in your data each combination occurs
3. Normalize those counts by the total data size in order to arrive a probabilities.

Note: The sum of joints must be equal to one

Learning a Joint Distribution

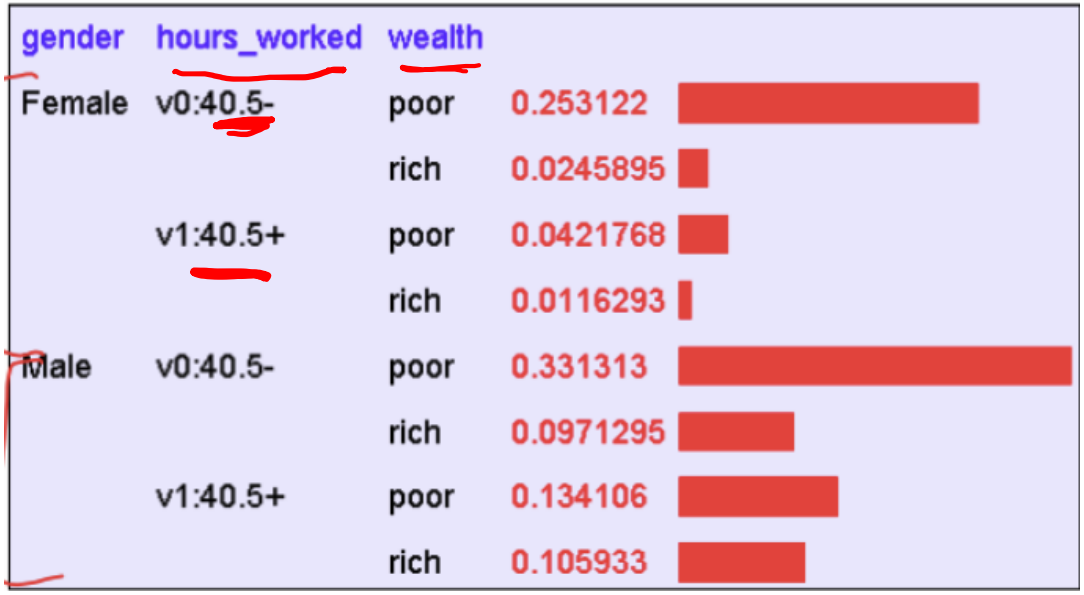
- To Build a JD (joint distribution) table for your attributes in which the probabilities are unspecified just fill in each row with

$$P(row) = \frac{records\ matching\ row}{total\ number\ of\ records}$$

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10

Example

- This JD was obtained by learning from three attributes in the UCI “Adult” Census database



gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

Using the Joint

- Once you have the JD you can easily compute the probability of any logical expression involving your attributes
- Using the *law of total probability* we can compute $P(Y)$ as the sum of all probabilities jointed with Y:

$$P(Y) = \sum_i P(Y \cap x_i) = \sum_{rows\ with\ Y} P(row)$$

- Examples:
  - What is P(Poor)?
  - What is P(Poor Male)?

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

Inference with the Joint

- As mentioned, now we can also easily compute joint/conditional probabilities using our definition of the joint:

$$P(y|x) = \frac{P(y \wedge x)}{P(x)} = \frac{\sum_{rows\ with\ y\ and\ x} P(row)}{\sum_{rows\ with\ x} P(row)}$$

- What is $P(Male|Poor)$?

$$P(M|P) = \frac{P(M,P)}{P(P)}$$

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

$$\frac{0.33 + 0.13}{0.46 + 0.04 + 0.25} = \frac{0.46}{0.75}$$

$P(R|F,-) = \frac{P(R,F,-)}{P(F,-)}$ $\frac{0.024}{0.024+0.253} = 0.086$

Example

- Suppose we want to figure out given gender and hours worked, what is the wealth?
  - We can also write this as $P(W|G,H)$
- What is $P(W = rich|G = female, H = 40.5-)$?

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

Inference is a big deal!

- There's tons of times you use it:
  - I've got this evidence. What's the chance that my conclusion is true?
  - I've got a sore neck. How likely am I to have Meningitis?
  - The lights are out and it's 9pm. What is the likelihood that my spouse is asleep?

Using Inference for Classification

- How can we use this for classification?
- Consider x, a set of D features
- To figure out which class a set of features should belong to we can just choose the class that maximizes the posterior probability

$$\hat{y} = \underset{i}{argmax}\, P(y = i | f = x)$$

$$= \underset{i}{argmax} \left(\frac{P(y = i, f = x)}{P(f = x)} \right)$$

$\frac{P(dog, pixels)}{P(pixels)}$

$P(cat, pixels)$ / $P(pixels)$

$P(mouse | pixels)$ / $P(pixels)$

Using Inference for Classification

$$\hat{y} = \underset{i}{argmax}\left(\frac{P(y = i, f = x)}{P(f = x)}\right)$$

- But since $P(f = x)$ is the same for all classes we can just do:

$$\hat{y} = \underset{i}{argmax}\, P(y = i, f = x)$$

- And if we have $P(y = i, f = x)$ for all classes i then you can compute the actual probabilities, $P(y = i | f = x)$ by dividing by the sum of the joint probabilities:

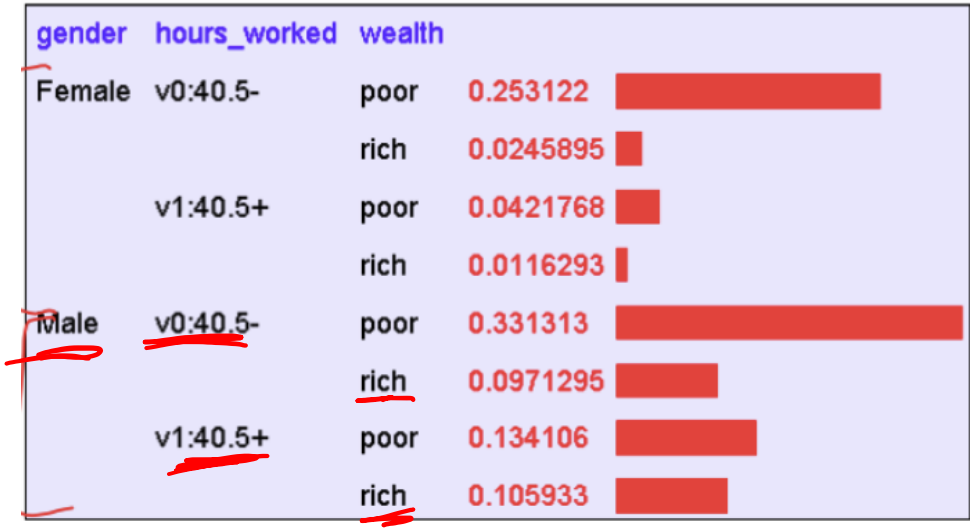
$$P(y = i|x) = \frac{P(y = i, f = x)}{\sum_j P(y = j, f = x)}$$

Inference for Classification Example

- Given a rich male let's classify them as having worked more or less than 40.5 hours per week
  - $P(40.5+|male, rich) \propto P(40.5+, male, rich)$
  - $P(40.5-|male, rich) \propto P(40.5-, male, rich)$

0.105

0.097



$$\frac{0.105}{0.105 + 0.097}$$