

CS 360: Machine Learning

Prof. Sara Mathieson

Fall 2019

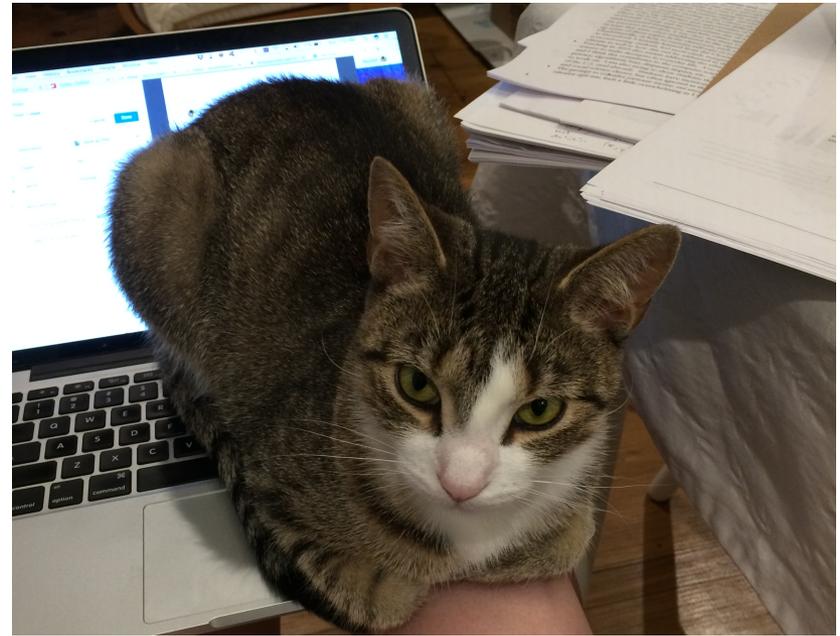
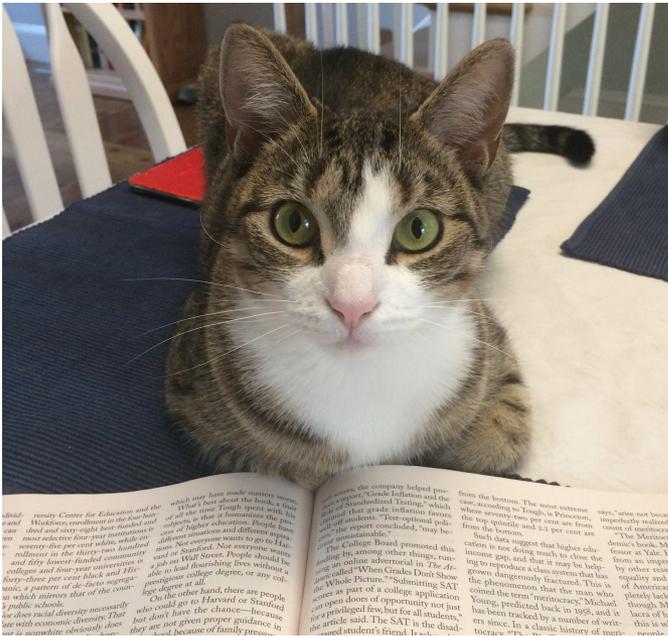


Haverford
COLLEGE

- **Lab 5 TODAY!**
 - Office hours today 12:30—1:30pm (H110)
- **Reading Quiz Thursday** (Duame Section 13.1)
- Lab 6 due Friday Nov 1
 - Checkpoint during lab on Thursday Oct 31 (Part 1 and 2)

In lab Thursday

- Hand back the midterm
- Go over common issues
- Start Lab 6



Outline for October 22

- Evaluation metrics
 - Confusion matrices revisited
 - ROC curves
 - Relationship to probabilistic methods
- Ensemble methods
 - Bagging
 - Random forests

Outline for October 22

- Evaluation metrics
 - Confusion matrices revisited
 - ROC curves
 - Relationship to probabilistic methods
- Ensemble methods
 - Bagging
 - Random forests

For now: assume binary classification task

- Transactions that indicate credit card fraud
- Detecting which scans show tumors
- Prenatal test for Down's Syndrome
- Finding genes under natural selection
- Finding regions of the genome with high recombination rate ("hotspots")

For now: assume binary classification task

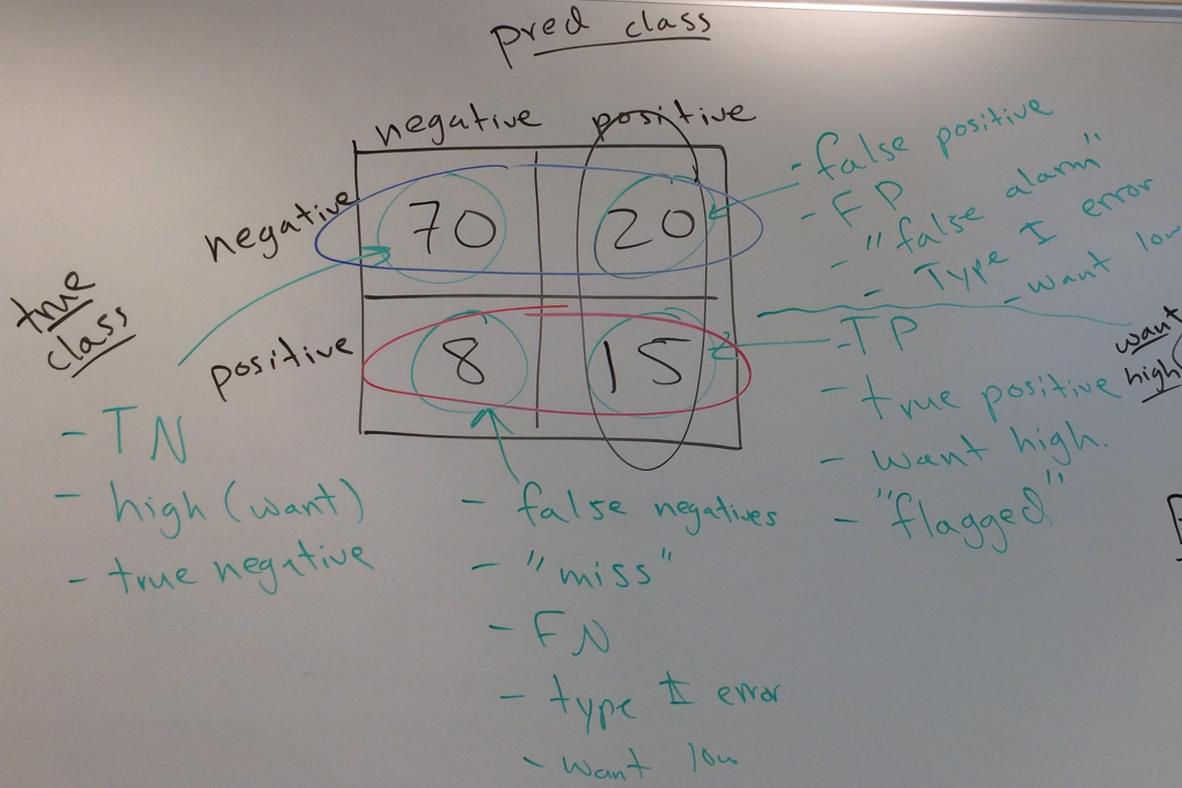
- Transactions that indicate credit card fraud
- Detecting which scans show tumors
- Prenatal test for Down's Syndrome
- Finding genes under natural selection
- Finding regions of the genome with high recombination rate ("hotspots")

In all these examples, we are trying to find unusual items ("needle in a haystack") -- we call these *positives*

Goals of Evaluation

- Think about what metrics are important for the problem at hand
- Compare different methods on the same problem
- Common set of tools that other researchers/users can understand

Back to Confusion Matrices...



Recall: how many positives were found?

true positive rate

$$TPR = \frac{TP}{FN+TP}$$

$$= \frac{15}{8+15} \approx 0.65$$

Precision:

$$= \frac{TP}{FP+TP}$$

$$= \frac{15}{20+15} \approx 0.43$$

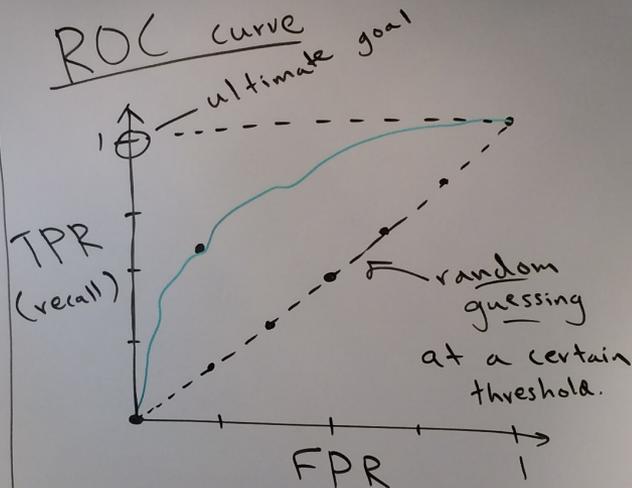
false positive rate

$$FPR = \frac{FP}{FP + TN}$$

want
low

$$= \frac{20}{70 + 20} \approx 0.22$$

ROC curve



0	90
0	23

90 = N

23 = P

$N^* = 0$ $P^* = 113$

predict all positive

$$FPR = \frac{90}{0 + 90} = 1$$

$$TPR = \frac{23}{0 + 23} = 1$$

90	0
23	0

FPR = 0

TPR = 0

predict all negative.

45	45
12	11

$$TPR = \frac{11}{23} \approx 0.5$$

$$FPR = \frac{45}{90} = 0.5$$

Probabilistic Model

threshold: 0.25 } only at
test time

$$p(y=1|x) \begin{cases} > 0.25 \Rightarrow \hat{y} = 1 \\ \leq 0.25 \Rightarrow \hat{y} = 0 \end{cases}$$

threshold: 0.75

want to be confident

Handout 12

①

	N	P
N	77	3
P	13	7

$N = 80$ ②
 $P = 20$
 $N^* = 90$ $P^* = 10$

precision

$$= \frac{7}{3+7}$$

$$= 0.70$$

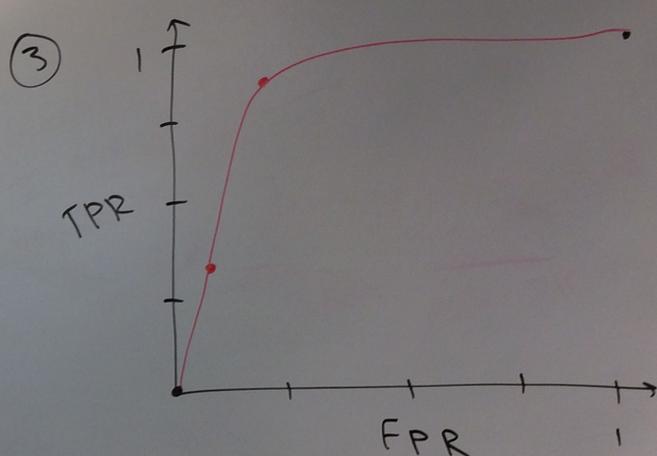
recall (TPR)

$$= \frac{7}{13+7}$$

$$= 0.35$$

FPR

$$= \frac{3}{80} = 0.04$$



68	12
2	18

$FPR = \frac{12}{80} \approx .15$
 $TPR = \frac{18}{20} = 0.9$

Precision and Recall

- Precision: of all the “flagged” examples, which ones are actually relevant (i.e. positive)?
- Recall: of all the relevant results, which ones did I actually return?

Precision and Recall

- Precision: of all the “flagged” examples, which ones are actually relevant (i.e. positive)?

(Purity)

- Recall: of all the relevant results, which ones did I actually return?

(Completeness)

Precision and Recall

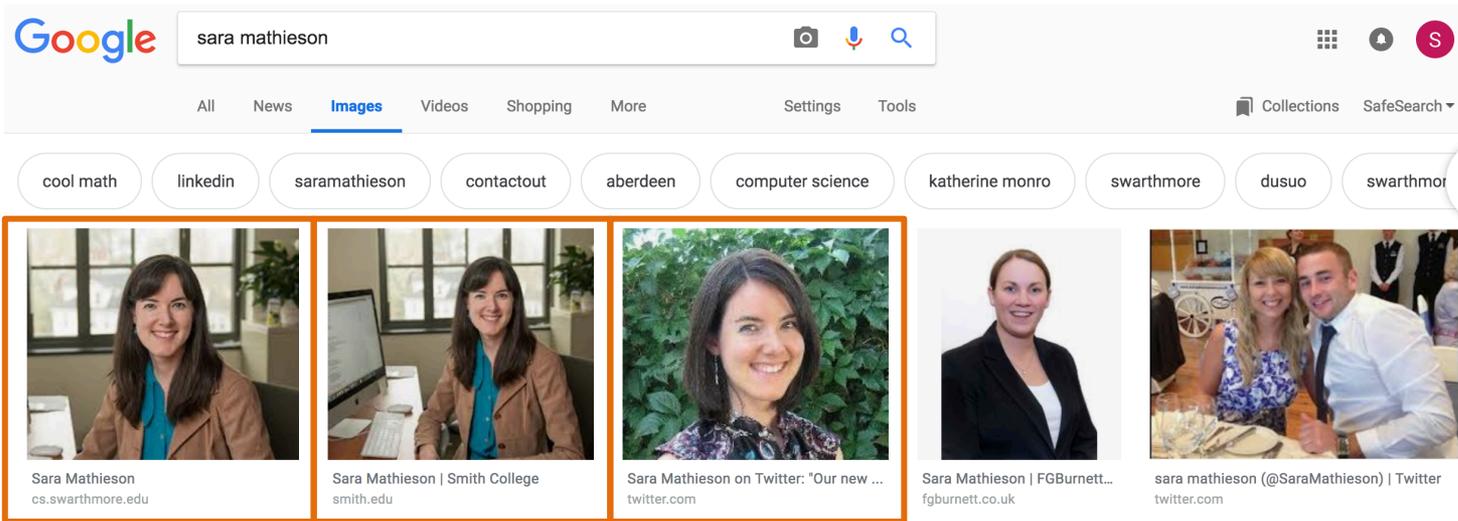
The screenshot shows a Google search for "sara mathieson" with the "Images" tab selected. The search bar contains "sara mathieson" and the Google logo is on the left. Below the search bar are navigation tabs for "All", "News", "Images", "Videos", "Shopping", and "More". To the right are "Settings" and "Tools". Further right are "Collections" and "SafeSearch". Below the navigation are several filter buttons: "cool math", "linkedin", "saramathieson", "contactout", "aberdeen", "computer science", "katherine monro", "swarthmore", "dusuo", and "swarthmor". The search results are displayed in a grid of five images. Each image has a caption below it:

- Image 1: Sara Mathieson, cs.swarthmore.edu
- Image 2: Sara Mathieson | Smith College, smith.edu
- Image 3: Sara Mathieson on Twitter: "Our new ...", twitter.com
- Image 4: Sara Mathieson | FGBurnett..., fgburnett.co.uk
- Image 5: sara mathieson (@SaraMathieson) | Twitter, twitter.com

$P=6$ (number of images that are actually me)

- Precision?
- Recall?

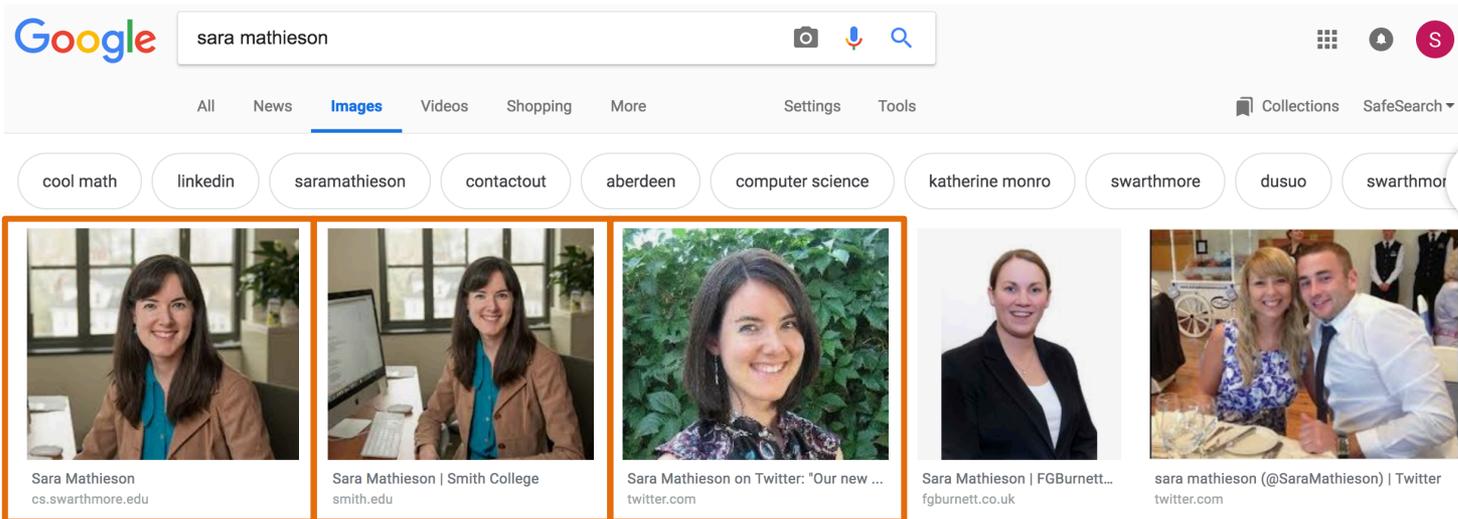
Precision and Recall



$P=6$ (number of images that are actually me)

- Precision = $TP/(FP+TP) = 3/5$
- Recall?

Precision and Recall



$P=6$ (number of images that are actually me)

- Precision = $TP/(FP+TP) = 3/5$
- Recall = $TP/(FN+TP) = 3/6$

Precision and Recall

The screenshot shows a Google Images search for "sara mathieson". The search results are displayed in a grid. Six results are highlighted with orange borders, indicating they are true positives. The highlighted results are:

- Sara Mathieson, cs.swarthmore.edu
- Sara Mathieson | Smith College, smith.edu
- Sara Mathieson on Twitter: "Our new ...", twitter.com
- Sara Mathieson - Graham + Si..., g-s.co.uk
- Sara Mathieson (Saramathi...), github.com
- Sara Mathieson Email &..., contactout.com

The other 10 results are false positives, including images of other people and Sara Mathieson in different contexts.

$P=6$ (number of images that are actually me)

- Precision = $5/16$
- Recall = $5/6$

Recap Confusion Matrices

Predicted class

Negative

Positive

Negative

True negative
(TN)

False positive
(FP)

True
class

Positive

False negative
(FN)

True positive
(TP)

Recap Confusion Matrices

Predicted class

Negative

Positive

Negative

True negative (TN)	False positive (FP) "false alarm"
False negative (FN) "miss"	True positive (TP)

N (total number of true negatives)

True class

Positive

P (total number of true positives)

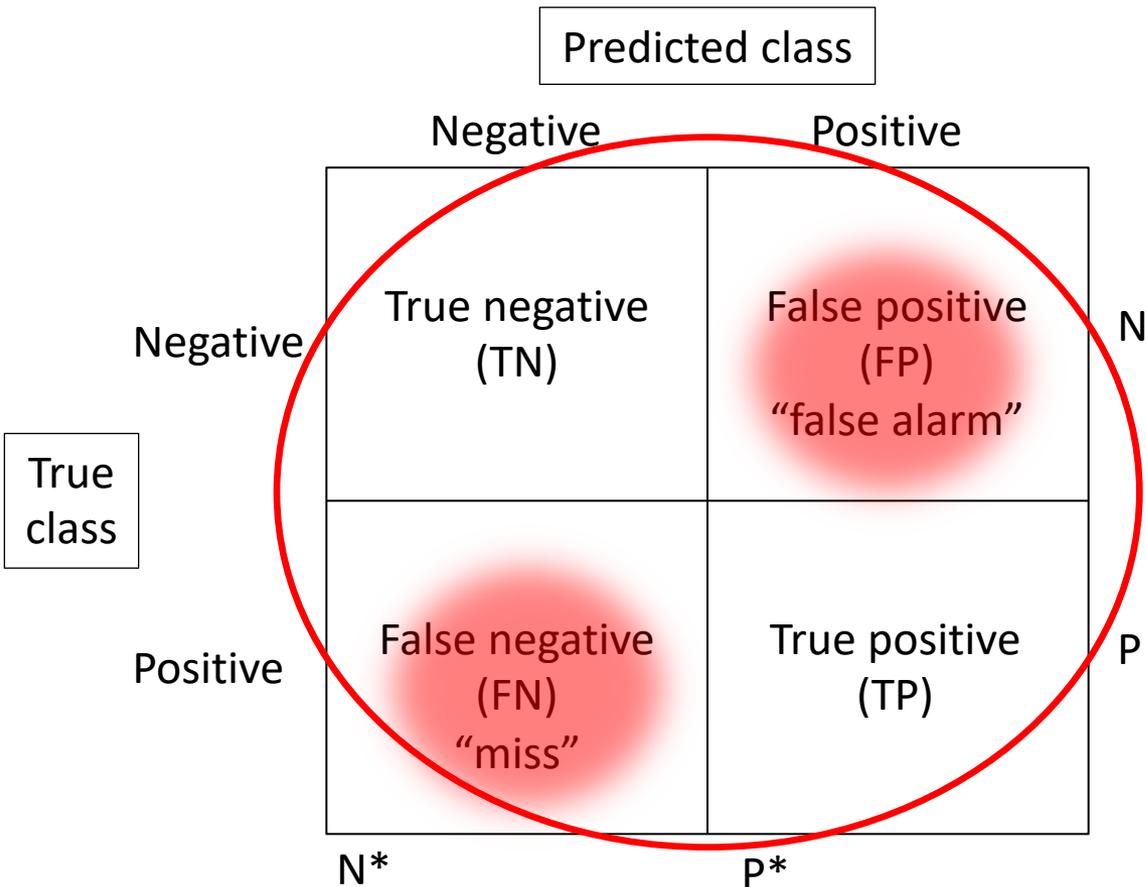
N* (what we said was negative)

P* (what we said was positive "flagged")

Recap Confusion Matrices

		Predicted class	
		Negative	Positive
True class	Negative	True negative (TN) ✓	False positive (FP) "false alarm" ✗
	Positive	False negative (FN) "miss" ✗	True positive (TP) ✓
		N*	p*

Recap Confusion Matrices

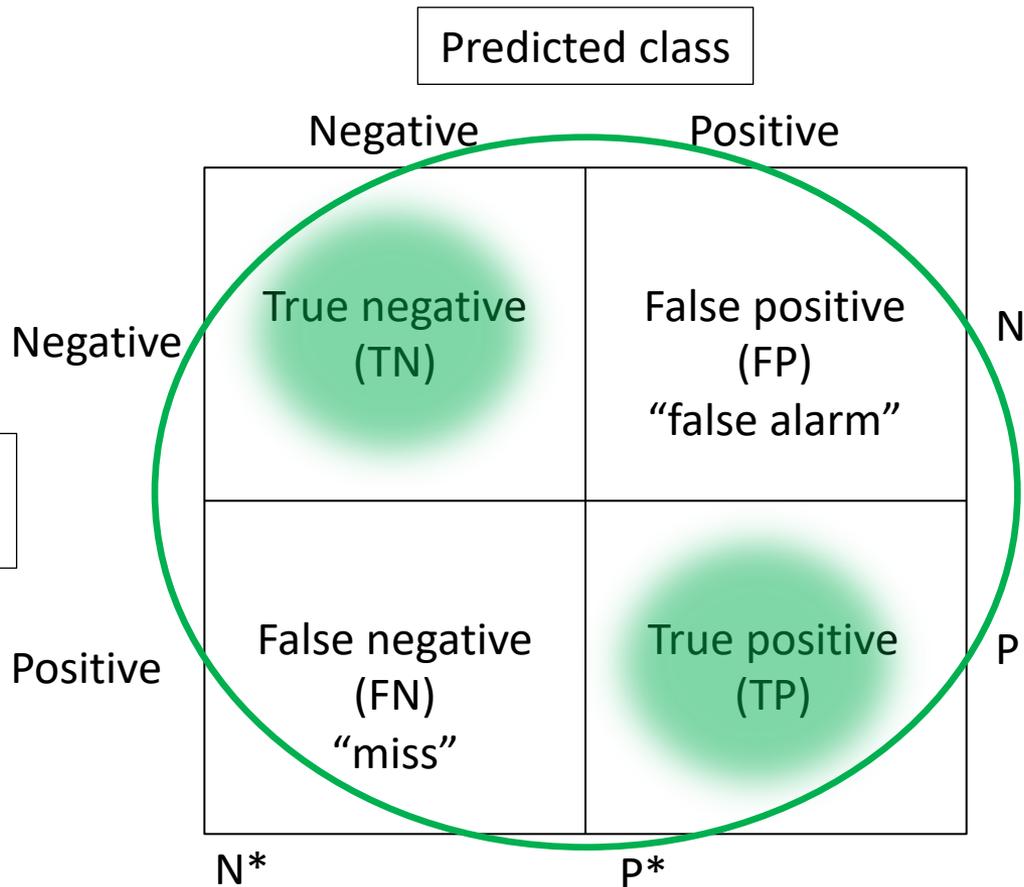


Error:

$$(FN+FP)/(TN+FP+FN+TP)$$

$$= (FN+FP)/(N+P)$$

Recap Confusion Matrices



Accuracy = 1-Error:

$$(TN+TP)/(TN+FP+FN+TP)$$

$$= (TN+TP)/(N+P)$$

Recap Confusion Matrices

		Predicted class	
		Negative	Positive
True class	Negative	True negative (TN)	False positive (FP) "false alarm"
	Positive	False negative (FN) "miss"	True positive (TP)
		N*	p*

The diagram shows a 2x2 confusion matrix. The columns are labeled 'Negative' and 'Positive' under the heading 'Predicted class'. The rows are labeled 'Negative' and 'Positive' under the heading 'True class'. The cells contain: True negative (TN), False positive (FP) "false alarm", False negative (FN) "miss", and True positive (TP). A purple oval highlights the FP and TP cells. The total number of negative predicted items is N, and the total number of positive predicted items is P. The total number of actual negative items is N*, and the total number of actual positive items is p*.

Precision:

$$TP / (FP + TP) = TP / P^*$$

Recap Confusion Matrices

		Predicted class	
		Negative	Positive
True class	Negative	True negative (TN)	False positive (FP) "false alarm"
	Positive	False negative (FN) "miss"	True positive (TP)
		N*	p*

The diagram shows a 2x2 confusion matrix. The top row is labeled 'Negative' and the bottom row is labeled 'Positive' under the 'True class' header. The left column is labeled 'Negative' and the right column is labeled 'Positive' under the 'Predicted class' header. The cells contain: True negative (TN), False positive (FP) "false alarm", False negative (FN) "miss", and True positive (TP). The TP cell is highlighted with a blue oval. Marginal counts N* and p* are shown at the bottom, and N and P are shown on the right side of the matrix.

Recall
(True Positive Rate):

$$TP/(FN+TP) = TP/P$$

Recap Confusion Matrices

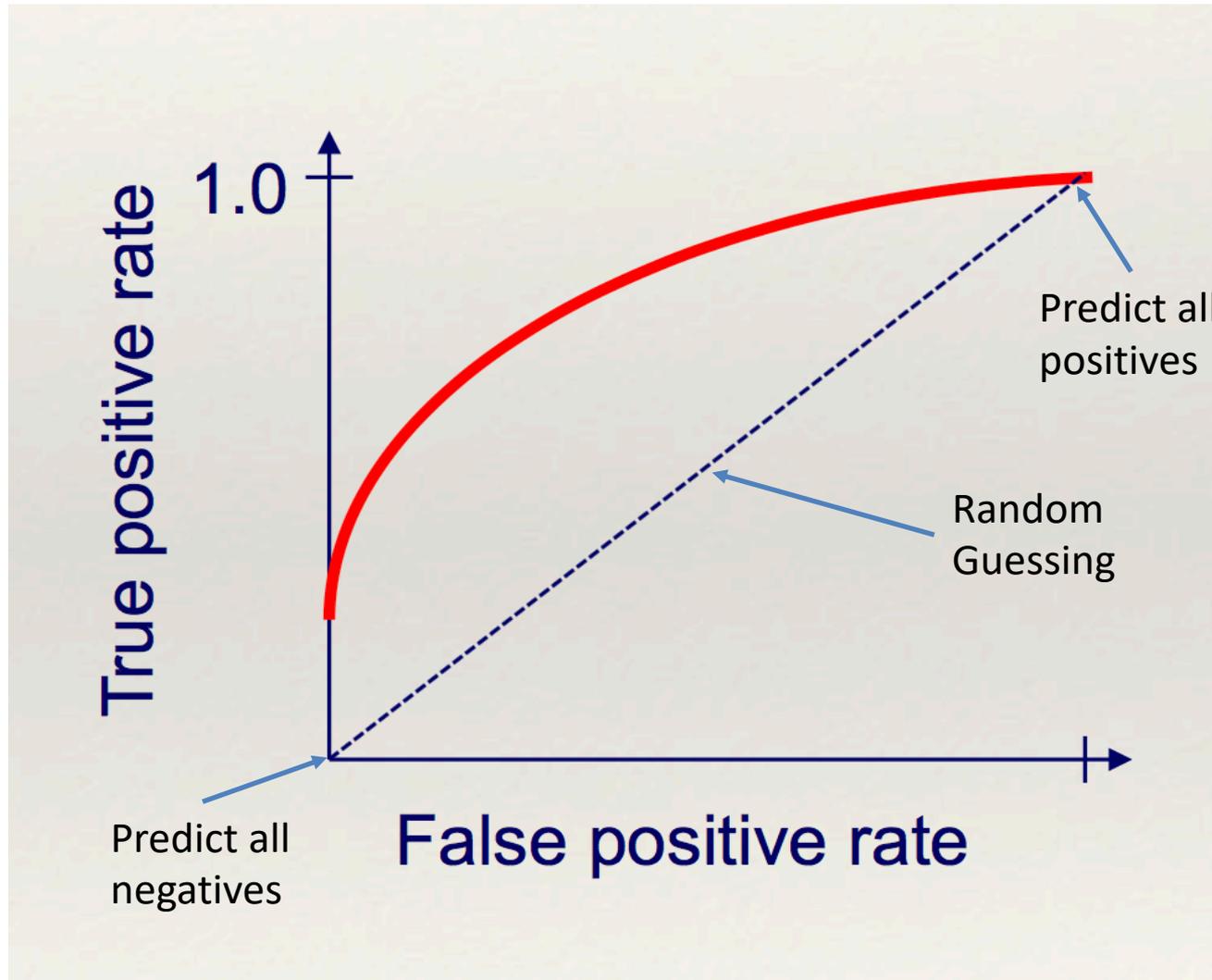
		Predicted class	
		Negative	Positive
True class	Negative	True negative (TN)	False positive (FP) "false alarm"
	Positive	False negative (FN) "miss"	True positive (TP)
		N*	p*

The diagram shows a 2x2 confusion matrix. The top row is labeled 'Negative' and the bottom row 'Positive' on the left side, representing the 'True class'. The top column is labeled 'Negative' and the bottom column 'Positive' on the top side, representing the 'Predicted class'. The four quadrants are: True negative (TN), False positive (FP) labeled as "false alarm", False negative (FN) labeled as "miss", and True positive (TP). The total number of negative true classes is N*, and the total number of positive true classes is p*. The total number of negative predicted classes is N, and the total number of positive predicted classes is P. An orange oval highlights the TN and FP cells, and the FP cell is shaded orange.

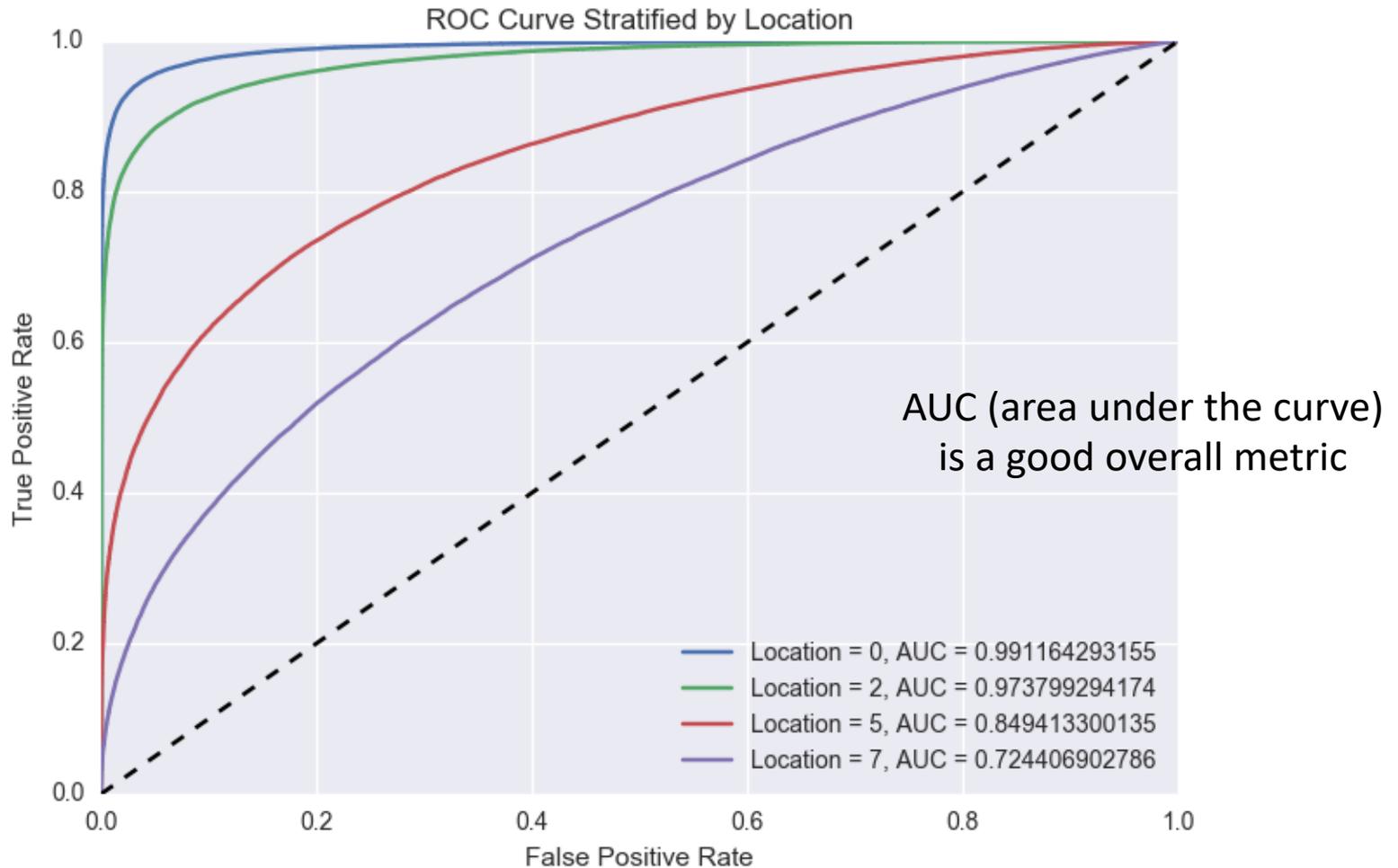
False Positive Rate:

$$FP/(TN+FP) = FP/N$$

ROC curve (Receiver Operating Characteristic)



ROC curve example: comparing methods



Example of a ROC curve from my research
Chan, Perrone, Spence, Jenkins, Mathieson, Song

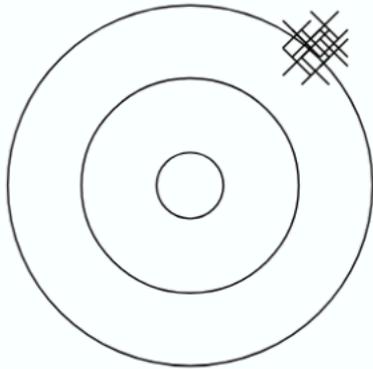
How to get a ROC curve for probabilistic methods?

- Usually we use 0.5 as a threshold for binary classification
- Vary the threshold! (i.e. choose 0.25)
 - $P(y=1 \mid x) > 0.25 \quad \Rightarrow$ classify as 1 (positive)
 - $P(y=1 \mid x) \leq 0.25 \quad \Rightarrow$ classify as 0 (negative)

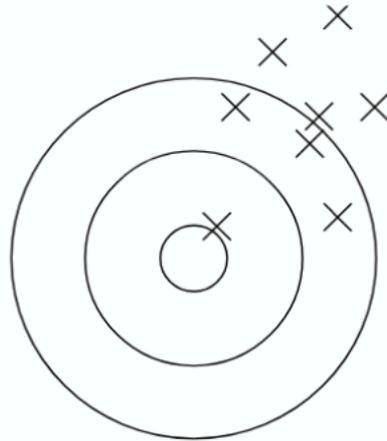
Outline for October 22

- Evaluation metrics
 - Confusion matrices revisited
 - ROC curves
 - Relationship to probabilistic methods
- Ensemble methods
 - Bagging
 - Random forests

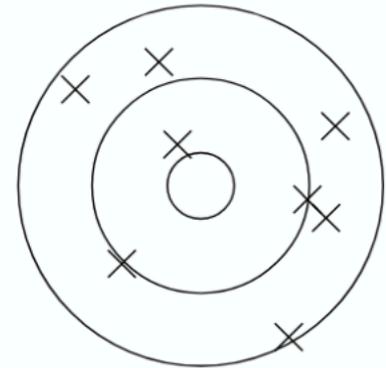
Quiz: recap bias and variance



A



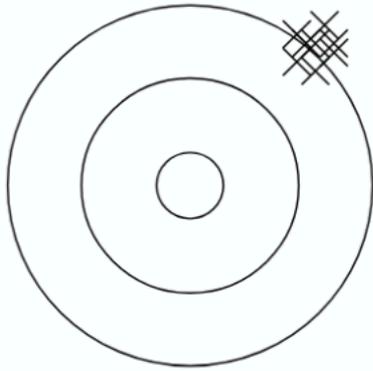
B



C

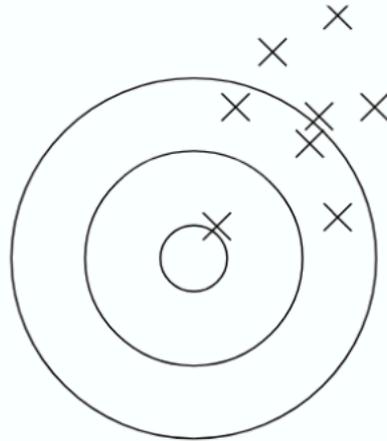
Label each picture with variance (high or low) and bias (high or low)

Quiz: recap bias and variance

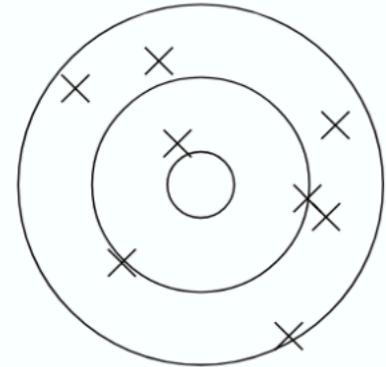


A

Variance: low
Bias: high



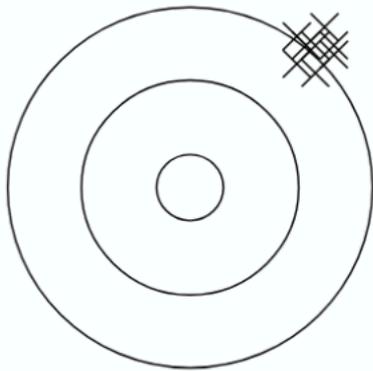
B



C

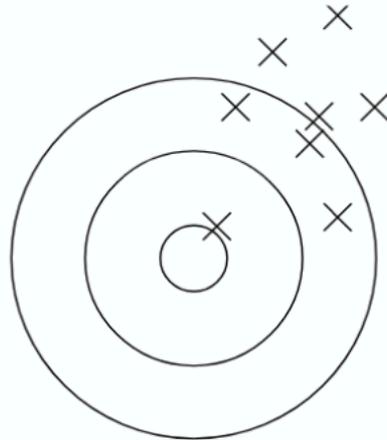
Label each picture with variance (high or low) and bias (high or low)

Quiz: recap bias and variance



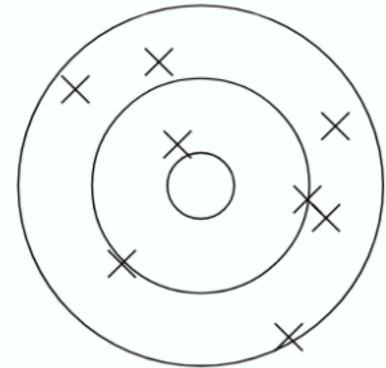
A

Variance: low
Bias: high



B

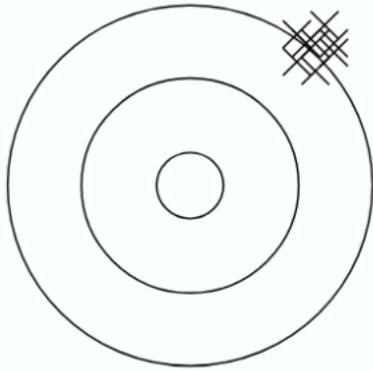
Variance: high
Bias: high



C

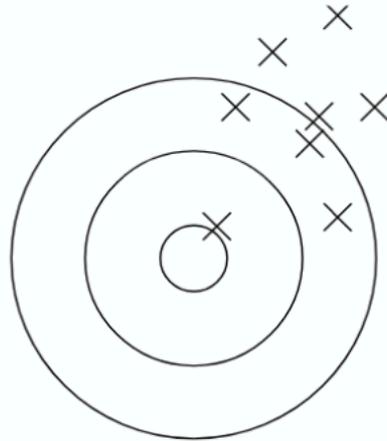
Label each picture with variance (high or low) and bias (high or low)

Quiz: recap bias and variance



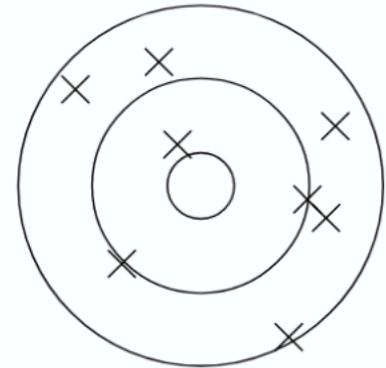
A

Variance: low
Bias: high



B

Variance: high
Bias: high

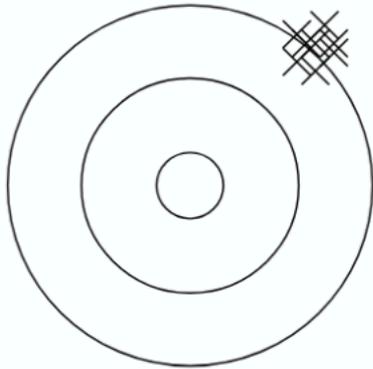


C

Variance: high
Bias: low

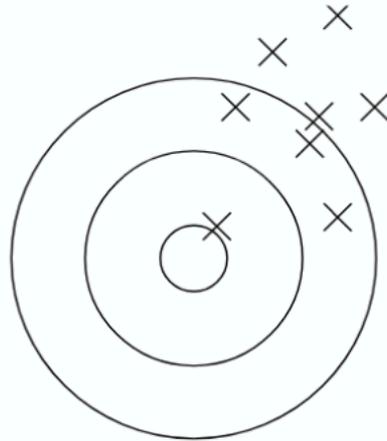
Label each picture with variance (high or low) and bias (high or low)

Quiz: recap bias and variance



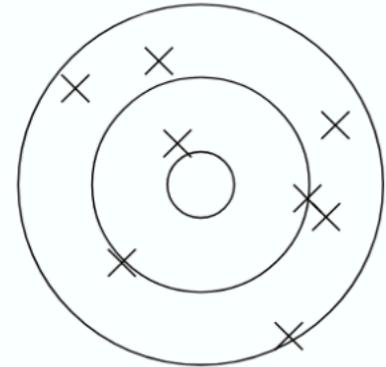
A

Variance: low
Bias: high



B

Variance: high
Bias: high



C

Variance: high
Bias: low

This is the type of classifier we want to average!

Label each picture with variance (high or low) and bias (high or low)

Ensemble Idea

- Average the results from several models with high variance and low bias
 - Important that models be diverse (don't want them to be wrong in the same ways)
- If n observations each have variance s^2 , then the mean of the observations has variance s^2/n (reduce variance by averaging!)

Learning Theory

Let H be the hypothesis space

Three sources of limitations for traditional classifiers:

- ❖ Statistical - H is too large relative to size of data
 - ❖ Many hypotheses can fit the data by chance
- ❖ Computational - H is too large to completely search for “best” model
- ❖ Representational - H is not expressive enough

Learning Theory

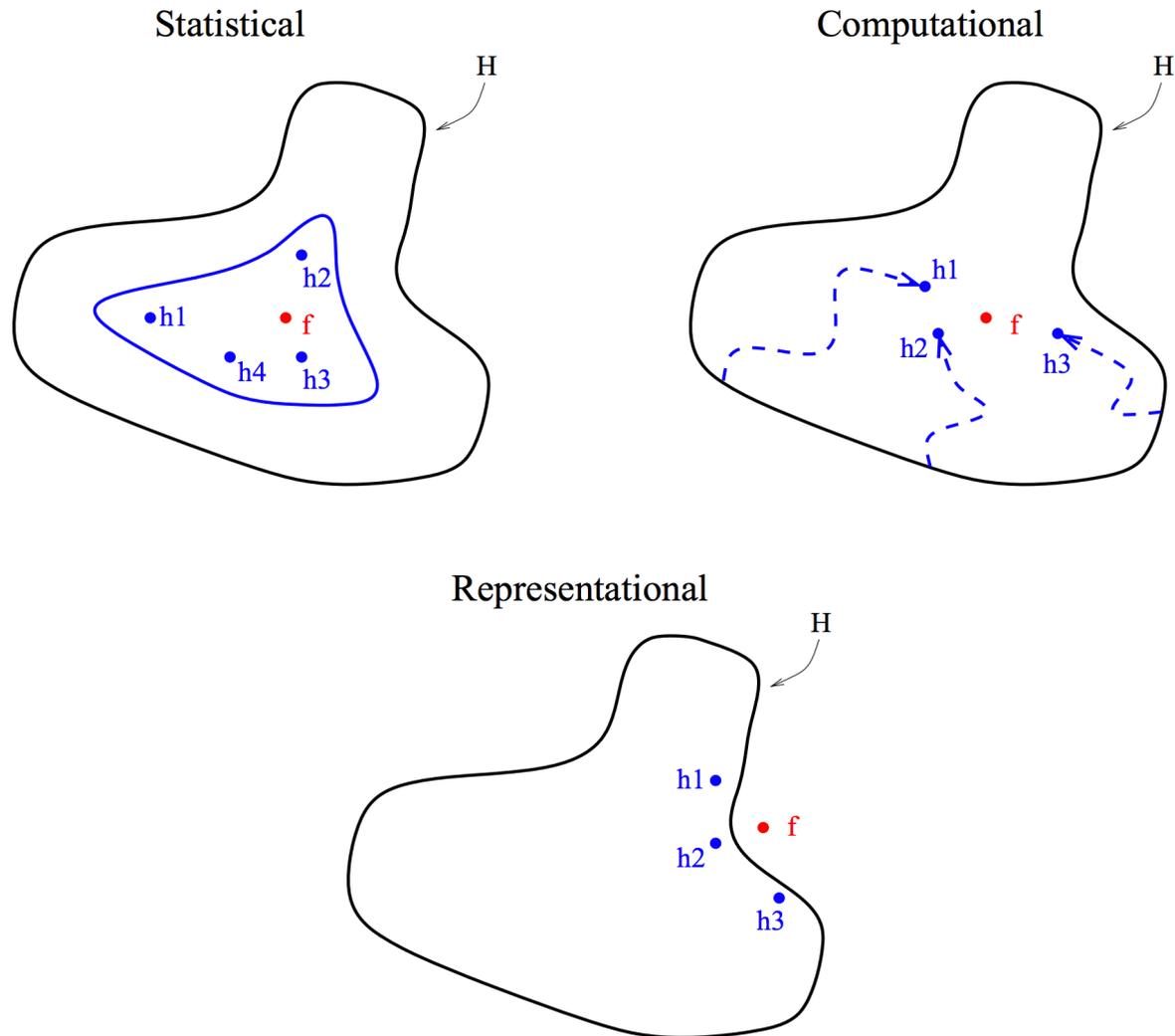
- ❖ Statistical: Average of unstable models (high variance) has more stability
- ❖ Computational: searching from multiple starting points is better approximation than one starting point
- ❖ Representational: sum of many models can represent more hypotheses than an individual model

Learning Theory

- ❖ Statistical: Average of unstable models (high variance) has more stability
- ❖ Computational: searching from multiple starting points is better approximation than one starting point
- ❖ Representational: sum of many models can represent more hypotheses than an individual model

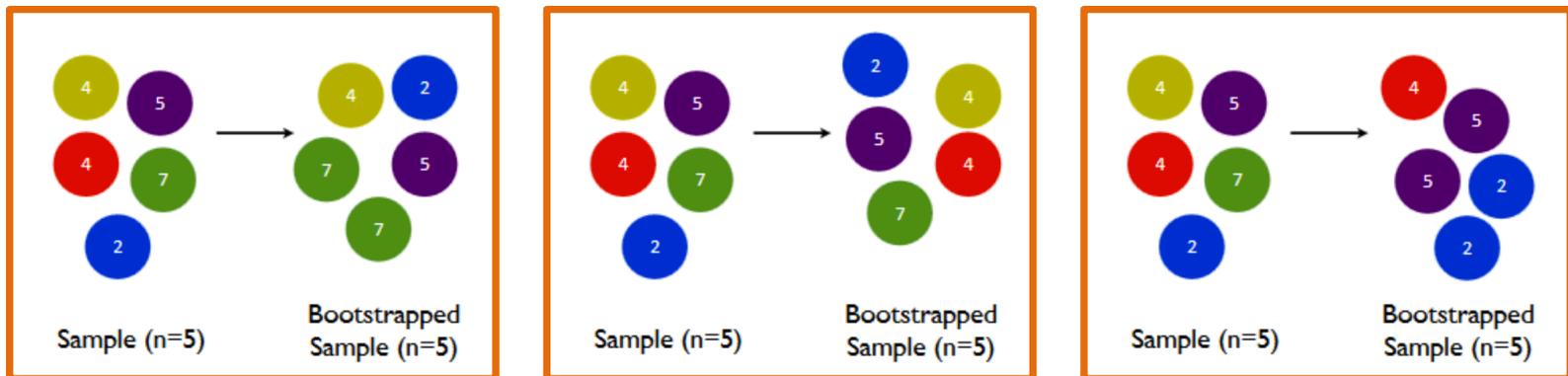
Ensembles can address all 3!

Learning Theory



Bagging Algorithm

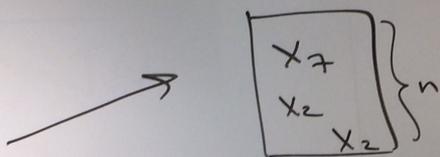
- ❖ Bagging = Bootstrap Aggregation [Brieman, 1996]
- ❖ *Bootstrap* (randomly sample with replacement) original data to create many different training sets
- ❖ Run base learning algorithm on each new data set independently



Desmond Ong, Stanford

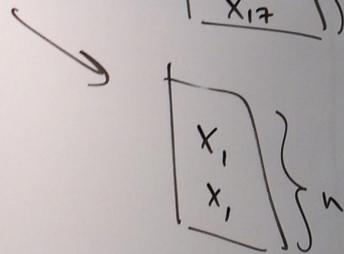
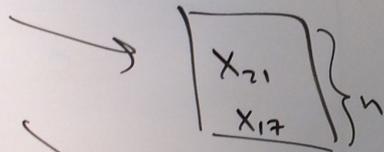
Bootstrap

sample with replacement



prob we don't
choose an example

$$\left(\frac{n-1}{n}\right)^n = \left(1 - \frac{1}{n}\right)^n \xrightarrow{\lim_{n \rightarrow \infty}} e^{-1} \approx \frac{0.38}{\text{large}}$$



$$\lim_{n \rightarrow \infty} \left(1 + \frac{x}{n}\right)^n = e^x$$

Notation

For Ensembles

$T = \# \text{ models/classifiers (index } t)$

$x = \text{ test example (could be vector)}$

$X^{(t)} = \text{ bootstrap training set } t$

$h^{(t)}(x) = \text{ hypothesis about } x$
from model t

$r = \text{ prob. of error of individual}$
model

$R = \# \text{ votes for wrong class.}$

Bagging (Bootstrap Aggregation), $y \in \{0, 1\}$

Train for t in range(T):

- create bootstrap dataset $X^{(t)}$ ($n \times p$)
- train on $X^{(t)}$ to get model $h^{(t)}$

also in $\{0, 1\}$
threshold
already applied

Test for x in test data:

$$h(x) = \operatorname{argmax}_{y \in \{0, 1\}} \sum_{t=1}^T \mathbb{I}(h^{(t)}(x) = y)$$