



# PERFORMANCE EVALUATION

CS6140

Predrag Radivojac

KHOURY COLLEGE OF COMPUTER SCIENCES  
NORTHEASTERN UNIVERSITY

Spring 2021

# NEED FOR (DEEP) EVALUATION IN MACHINE LEARNING

A turtle—or a rifle? Hackers easily fool AIs into seeing the wrong thing

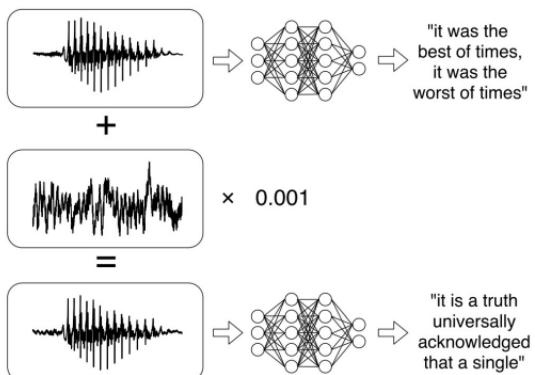
By Matthew Hutson | Jul. 19, 2018, 2:15 PM



## Artificial Intelligence Is Rushing Into Patient Care—And Could Raise Risks

AI systems are not as rigorously tested as other medical devices, and have already made serious mistakes

### Audio Adversarial Examples: Targeted Attacks on Speech-to-Text



With the help of stickers, image recognition algorithms were tricked into thinking a stop sign was a speed limit sign. K. EYKHOLT ET AL.; ARXIV:1707.08945 (2017)

Abstract | PDF | Request permissions



Figure 1. Illustration of our attack: given any waveform, adding a small perturbation makes the result transcribe as any desired target phrase.

# EVALUATION IN MACHINE LEARNING

## Machine learning:

- field with strong theory, rooted in probability, statistics, computer science
- algorithms and frameworks must work on data from the real world
- algorithms and models must be evaluated on data from the real world
- need mechanisms to find the best model or algorithm out of many options

## Empirical evaluation:

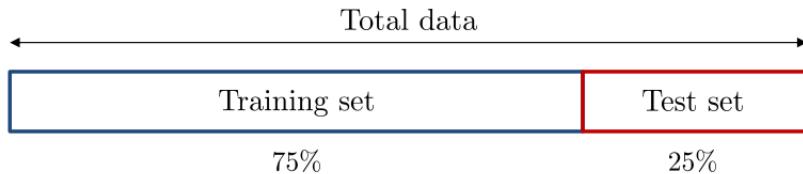
- core part of machine learning
- gives confidence to deploy
- our focus on accuracy estimation

## Related important concepts:

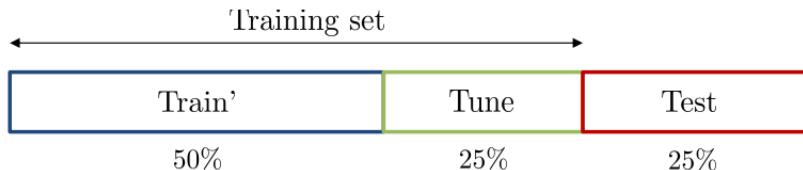
- reliability
- interpretability
- fairness

## BASIC IDEA FOR THE PROTOCOL (ALL DATA LABELED)

- randomly split data into training set and test set (i.i.d.)
- train a model on the training set; predict on the test set and check accuracy



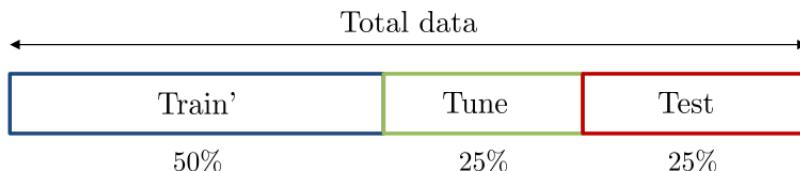
- works when we train a single model and have a lot of data
- what if we have to pick between a few models?



Tune set = Development set

# How DOES THE PROCESS WORK?

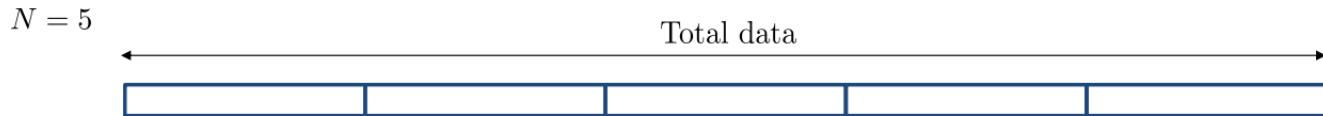
- train many models on Train' (e.g., vary parameters)
- pick the best parameters using Tune based on some criterion ← model selection
- train a model with selected parameters on Train = Train' + Tune
- compute accuracy on Test ← estimated performance
- train the final model on all data (+ model selection) ← final model
- deploy the final model with the hope it works as well as we think



Test set can be used only once, at the very end.

# CROSS-VALIDATION

- data less abundant relative to the complexity of model space
- partition data into  $N$  groups (uniformly randomly)



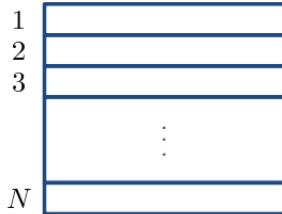
Repeat  $N$  times

- (step  $i$ ) Test = partition  $i$ ; Training = the rest
- (step  $i$ ) Compute accuracy of step  $i$

Average  $N$  accuracy measures at the end



# N-FOLD CROSS-VALIDATION



$f^{-i}$  = model trained on partitions  $\{1, 2, \dots, N\} \setminus \{i\}$

$\text{perf}^{-i}$  = performance of  $f^{-i}$  on test partition  $i$

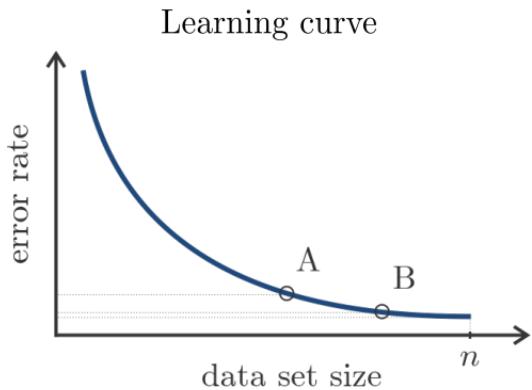
$\text{perf}$  = final performance, averaged over  $N$  different  $\text{perf}^{-i}$ 's

## Two questions:

- Is it a problem that each  $f^{-i}$  is different? No.
- How should we choose  $N$ ?

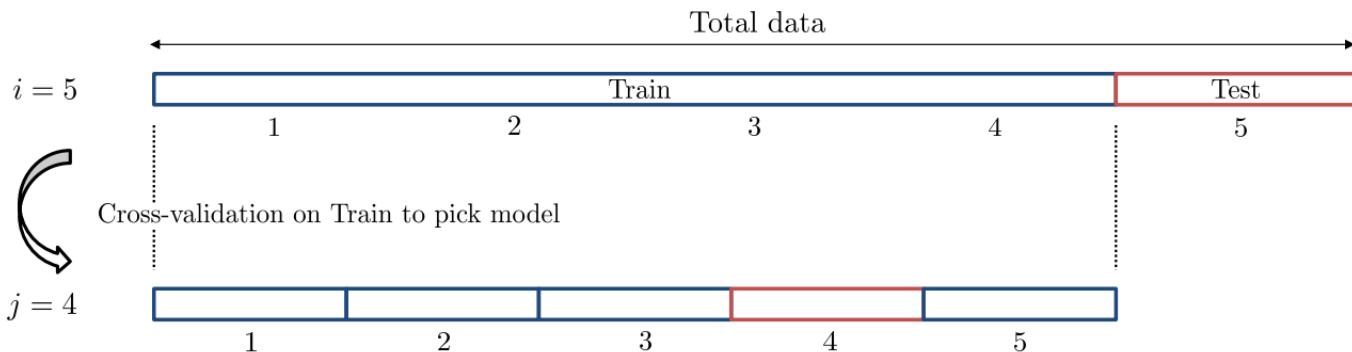
## Choice of $N$ :

- point A: high bias, low variance
- point B: low bias, high variance
- often  $N = 10$ , giving 10-fold cross-validation
- when  $N = n$ , it is called leave-one-out estimation



# NESTED CROSS-VALIDATION

- old: Train = Train' + Tune → best model
- new: N-fold cross-validation on Train → best model



## Run time:

- each parameter combination is trained  $N^2$  times;  $\forall i, j \in \{1, 2, \dots, N\}$
- best models are trained  $N$  times combined

# THE ROLES OF THE TRAINING DATA

## Estimate accuracy:

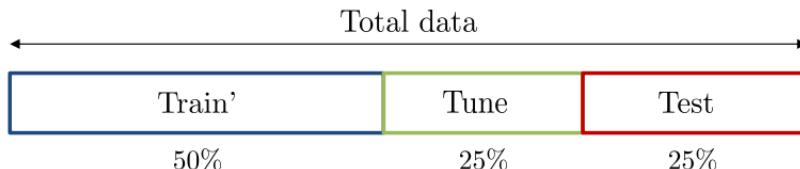
- nested cross-validation on the entire data set
- this accuracy is reported for the model trained at the very end

## Choose parameters:

- cross-validation on the entire data set
- the accuracy of this model may not match the estimated accuracy

## Train final model:

- retrain the best performing model on the entire data
- this model is never really evaluated (unless new data accumulates)



## OTHER EVALUATION SCENARIOS

### Random sampling:

- randomly sample w/o replacement a fraction of the data set
- train on training set, test on test set
- repeat many times

### Bootstrapping:

- sample w/ replacement  $n$  examples from the data set
- train on training set, test on examples not selected during sampling
- repeat many times

### Out-of-bag evaluation:

- applies to ensembles of models (committee machines)
- performs bootstrap-like estimation using multiple trained models

# CLASSIFICATION

**Given:** Test data:  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ ,  $x \in \mathcal{X}$  and  $y \in \mathcal{Y}$ , drawn i.i.d. from  $p(x, y)$

Trained model:  $f(x) \in \mathcal{Y}$

**Error rate and accuracy:**

$$\text{error} = P(f(X) \neq Y)$$

$$\text{accuracy} = 1 - \text{error}$$

**Assumption:**  $\mathcal{D}$  sampled i.i.d. from  $p(x, y)$

$$e_T = P(f(X) \neq Y) \quad \leftarrow \text{true error}$$

$$e_S = \frac{1}{n} \sum_{i=1}^n I(f(x_i) \neq y_i) \quad \leftarrow \text{estimated (sample) error}$$

**Confidence intervals:**

With 95% probability it holds that:

$$e_T = e_S \pm 1.96 \cdot \underbrace{\sqrt{\frac{e_S(1 - e_S)}{n}}}_{\text{standard error}}$$

Details later...

## BINARY CLASSIFICATION: METRICS

**Given:** Test data:  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ ,  $x \in \mathcal{X}$  and  $y \in \mathcal{Y} = \{0, 1\}$ , drawn i.i.d. from  $p(x, y)$

Trained model:  $f(x) \in \mathcal{Y}$

**True positive rate, sensitivity, recall:**  $P(f(X) = 1|Y = 1)$

**False positive rate:**  $P(f(X) = 1|Y = 0)$

**True negative rate, specificity:**  $P(f(X) = 0|Y = 0) = 1 - P(f(X) = 1|Y = 0)$

**False negative rate:**  $P(f(X) = 0|Y = 1) = 1 - P(f(X) = 1|Y = 1)$

**Precision, positive predictive value:**  $P(Y = 1|f(X) = 1)$

**False discovery rate:**  $P(Y = 0|f(X) = 1) = 1 - P(Y = 1|f(X) = 1)$

**Negative predictive value:**  $P(Y = 0|f(X) = 0)$

## BINARY CLASSIFICATION: DERIVED METRICS

**Given:** Test data:  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ ,  $x \in \mathcal{X}$  and  $y \in \mathcal{Y} = \{0, 1\}$ , drawn i.i.d. from  $p(x, y)$

Trained model:  $f(x) \in \mathcal{Y}$

**Balanced accuracy:** 
$$\frac{\text{sensitivity} + \text{specificity}}{2}$$

**Matthews correlation:**  $\text{Corr}[f(X), Y]$

**F-measure:** 
$$\frac{1}{\alpha \cdot \frac{1}{\text{precision}} + (1 - \alpha) \cdot \frac{1}{\text{recall}}} = \frac{(\beta^2 + 1) \cdot \text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$
  $\beta^2 = \frac{1-\alpha}{\alpha}$

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad \text{when } \alpha = \frac{1}{2}, \beta = 1$$

# ESTIMATION

Confusion matrix:

		Predicted class	
		positive	negative
True class	positive	tp	fn
	negative	fp	tn

tp = true positives

fp = false positives

tn = true negatives

fn = false negatives

$$tp + tn + fp + fn = n$$

Estimated quantities:

$$tpr = \frac{tp}{tp+fn}$$

$$\text{accuracy} = \frac{tp+tn}{n}$$

$$mcc = \frac{tp \cdot tn - fp \cdot fn}{\sqrt{(tp+fp)(tp+fn)(tn+fp)(tn+fn)}}$$

$$fpr = \frac{fp}{fp+tn}$$

$$\text{error} = \frac{fp+fn}{n}$$

$$fdr = \frac{fp}{fp+tp}$$

# ESTIMATION

Confusion matrix:

		Predicted class	
		positive	negative
True class	positive	tp	fn
	negative	fp	tn

tp = true positives

fp = false positives

tn = true negatives

fn = false negatives

$$tp + tn + fp + fn = n$$

Estimated quantities:

$$tpr = \frac{tp}{tp+fn}$$

$$\text{accuracy} = \frac{tp+tn}{n}$$

$$mcc = \frac{tp \cdot tn - fp \cdot fn}{\sqrt{(tp+fp)(tp+fn)(tn+fp)(tn+fn)}}$$

$$fpr = \frac{fp}{fp+tn}$$

$$\text{error} = \frac{fp+fn}{n}$$

$$fdr = \frac{fp}{fp+tp}$$

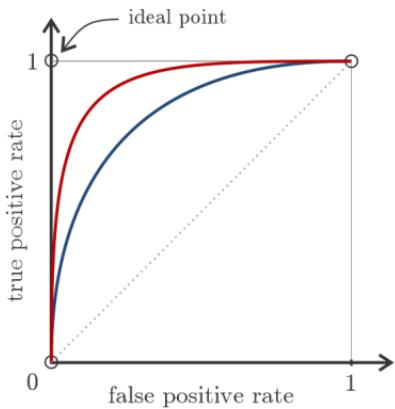
## TWO-DIMENSIONAL PLOTS

- a predictor is typically given as a score function  $s(x) \in \mathbb{R}$
- a binarized prediction  $f(x)$  is usually obtained by thresholding  $s(x)$  as

$$f_\tau(x) = \begin{cases} 1 & s(x) \geq \tau \\ 0 & s(x) < \tau \end{cases}$$

### Receiver Operating Characteristic (ROC) Curve:

- consider a series of thresholds  $\mathcal{T}$  and predictors  $f_\tau(x), \tau \in \mathcal{T}$
- each  $f_\tau(x)$  gives a true positive and a false positive rate
- ROC curve is a plot of  $\text{tpr}_\tau$  as a function of  $\text{fpr}_\tau$
- the red predictor is better than the blue; see Figure



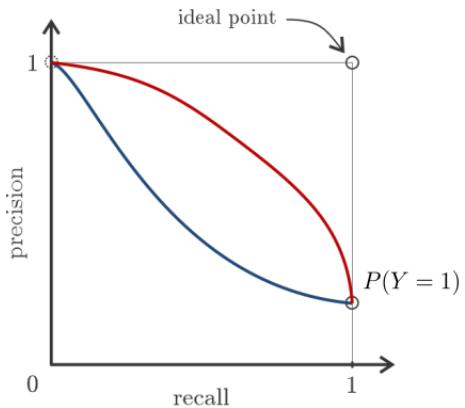
## TWO-DIMENSIONAL PLOTS

- a predictor is typically given as a score function  $s(x) \in \mathbb{R}$
- a binarized prediction  $f(x)$  is usually obtained by thresholding  $s(x)$  as

$$f_\tau(x) = \begin{cases} 1 & s(x) \geq \tau \\ 0 & s(x) < \tau \end{cases}$$

### Precision-Recall Curve:

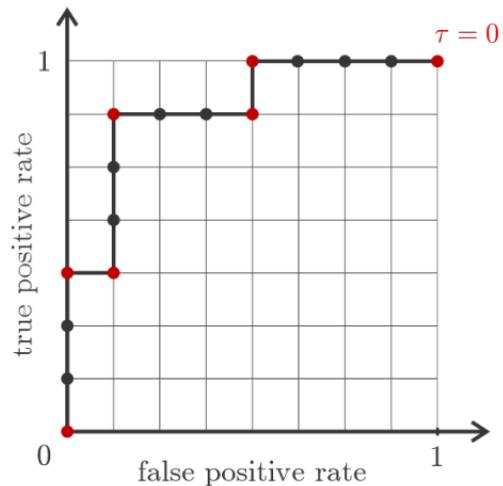
- consider a series of thresholds  $\mathcal{T}$  and predictors  $f_\tau(x), \tau \in \mathcal{T}$
- each  $f_\tau(x)$  gives a precision and a recall
- pr-rc curve is a plot of  $\text{pr}_\tau$  as a function of  $\text{rc}_\tau$
- the red predictor is better than the blue; see Figure



# AREA UNDER THE ROC CURVE (AUC)

- we have a data set with  $n = 15$  points and predictions  $s(x) \in [0, 1]$
- there are  $n_1 = 7$  positive and  $n_0 = 8$  negative examples

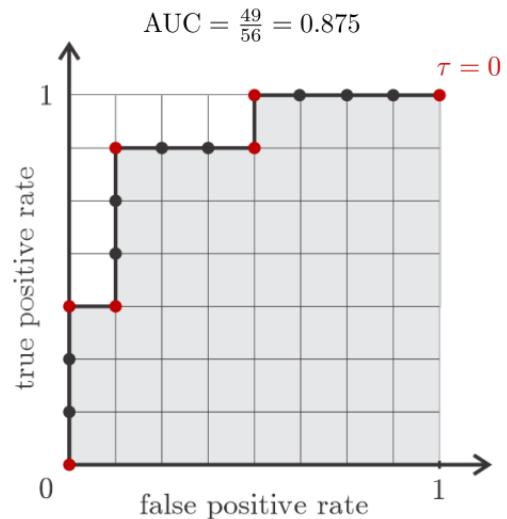
Prediction	True class
0.953	1
0.920	1
0.799	1
0.788	0
0.750	1
0.679	1
0.612	1
0.583	0
0.477	0
0.378	0
0.367	1
0.248	0
0.214	0
0.157	0
0.112	0



# AREA UNDER THE ROC CURVE (AUC)

- we have a data set with  $n = 15$  points and predictions  $s(x) \in [0, 1]$
- there are  $n_1 = 7$  positive and  $n_0 = 8$  negative examples

Prediction	True class
0.953	1
0.920	1
0.799	1
0.788	0
0.750	1
0.679	1
0.612	1
0.583	0
0.477	0
0.378	0
0.367	1
0.248	0
0.214	0
0.157	0
0.112	0



# AUC IS A RANKING MEASURE

- two models that provide the same ranking of examples in a data set have identical AUCs

Model 1

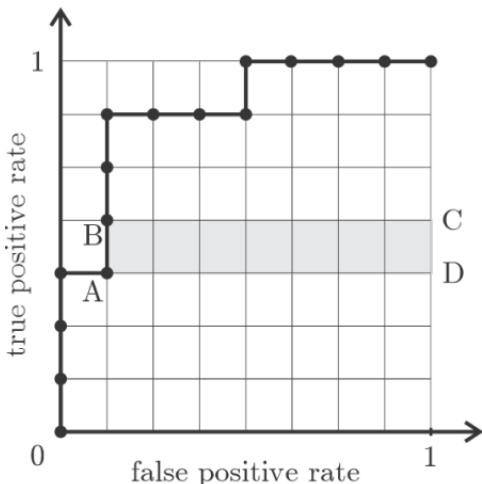
Example	Prediction	True class
6	0.953	1
9	0.920	1
5	0.799	1
13	0.788	0
14	0.750	1
12	0.679	1
7	0.612	1
10	0.583	0
8	0.477	0
11	0.378	0
2	0.367	1
4	0.248	0
1	0.214	0
3	0.157	0
15	0.112	0

Model 2

Example	Prediction	True class
6	1.000	1
9	0.999	1
5	0.998	1
13	0.909	0
14	0.895	1
12	0.888	1
7	0.845	1
10	0.844	0
8	0.830	0
11	0.828	0
2	0.818	1
4	0.817	0
1	0.816	0
3	0.814	0
15	0.800	0

# INTERPRETATION OF AUC

AUC = probability that a randomly chosen positive example has a higher score than a randomly chosen negative example



**Proof.\*** ABCD area =  $\frac{1}{n_1} \cdot \frac{n_0 - 1}{n_0}$

$f$  = number of false positives at present point

$$\text{AUC} \stackrel{\text{est}}{=} \frac{1}{n_1 n_0} \sum_{i=1}^{n_1} (n_0 - f_i)$$

$x^+$  = positive example;  $x^-$  = negative example;  $s(x)$  = prediction on  $x$

$$S(x^+, x^-) = \begin{cases} 1 & s(x^+) > s(x^-) \\ 0 & \text{otherwise} \end{cases}$$

$$P(s(x^+) > s(x^-)) \stackrel{\text{est}}{=} \frac{\sum_{i=1}^{n_1} \sum_{j=1}^{n_0} S(x_i^+, x_j^-)}{n_1 n_0} = \frac{1}{n_1 n_0} \sum_{i=1}^{n_1} (n_0 - f_i)$$

$$\text{because } \sum_{j=1}^{n_0} S(x_i^+, x_j^-) = n_0 - f_i$$

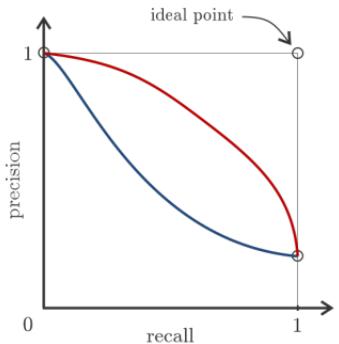
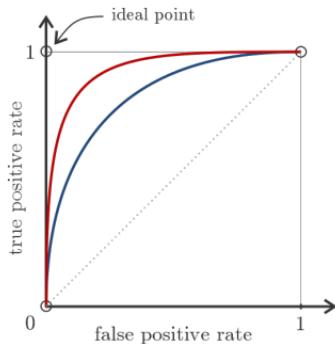
\* = no two data points have the same prediction score

# ROC CURVE VS. PR-RC CURVE

**Given:** Two models and a data set

**Claim 1:** If ROC curve 1 dominates ROC curve 2, then the corresponding pr-rc curve 1 dominates pr-rc curve 2.

**Claim 2:** Algorithms that optimize for area under the ROC curve  
do not necessarily optimize for the area  
under the pr-rc curve.



# TRIVIAL AND RANDOM CLASSIFIERS

Let  $p(y)$  be the distribution of class priors;  $y \in \mathcal{Y}$ .

**Trivial classifier:** predicts the most prevalent class from the training set.

$$\hat{y} = \arg \max_{y \in \mathcal{Y}} \{p(y)\} \quad \leftarrow \text{maximum a priori (instead of "a posteriori")}$$

**Random classifier:** Draw a random number from options in  $\mathcal{Y}$  according to  $p(y)$ .

**Example:** below is a confusion matrix for some classifier.

What is the accuracy of trivial and random classifier?

$$P(\text{corr}) = P(\text{corr}|+)P(+) + P(\text{corr}|-)P(-)$$

		Predicted class		
		positive	negative	
True class	positive	90	10	$p(Y = 0) = \frac{900}{1000}$
	negative	90	810	$p(Y = 1) = \frac{100}{1000}$

**Trivial:**  $\max\left(\frac{900}{1000}, \frac{100}{1000}\right) = 0.90$

**Random:**  $\left(\frac{100}{1000}\right)^2 + \left(\frac{900}{1000}\right)^2 = 0.82$

# ACCURACY OF TRIVIAL AND RANDOM CLASSIFIERS

**Trivial classifier:**

$$\text{accuracy} = \max_{y \in \mathcal{Y}} \{p(y)\}$$

**Random classifier:**

$$\text{accuracy} = \sum_{y \in \mathcal{Y}} p(y)^2 \quad 1 - \text{gini index} \text{ (used for classification trees)}$$

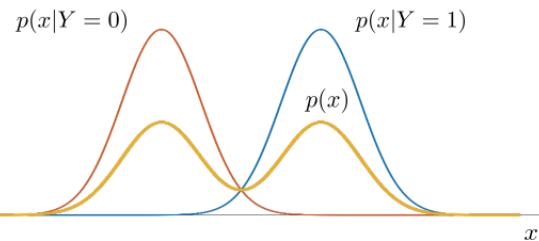
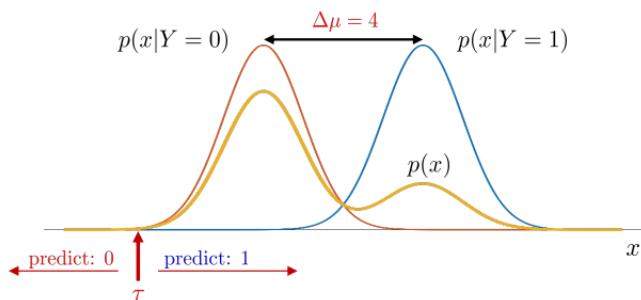
- these two are identical when  $p(y)$  is uniform
- estimates are found by estimating class priors  $p(y)$

**Relationship to decision tree splitting criteria:**

- minimum error: error of a trivial classifier
- gini index: error of a random classifier

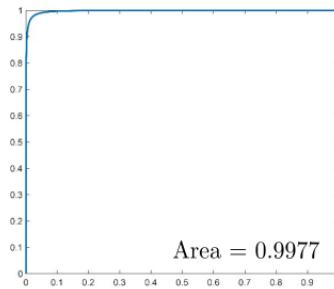
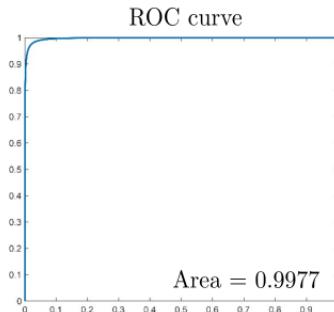
# IMBALANCED DATA: IMPACT ON 2D MEASURES

- class-conditional distributions are unit-variance Gaussians with  $\Delta\mu = 4$ ; predictor based on a threshold.
- imbalanced labeled data vs. balanced labeled data

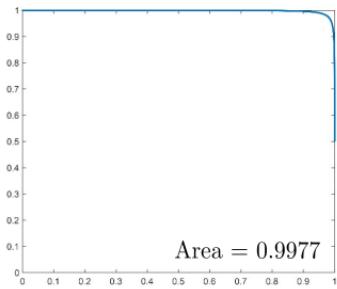
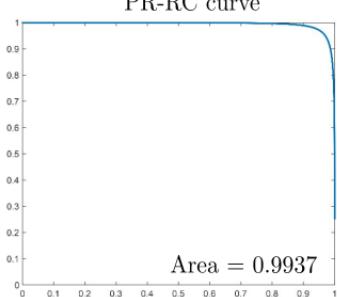


$$P(Y = 1) = \frac{1}{4}$$

$$P(Y = 1) = \frac{1}{2}$$



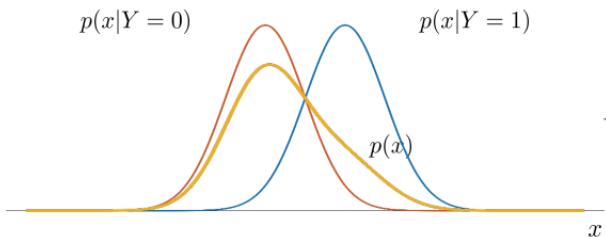
ROC curve



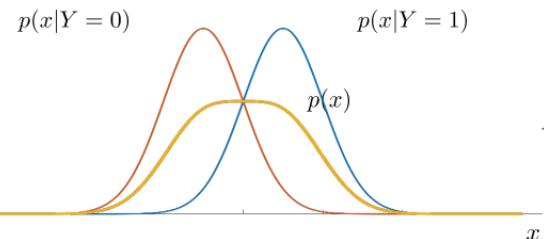
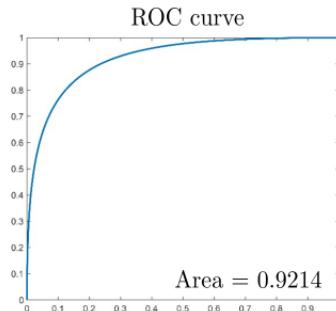
PR-RC curve

# IMBALANCED DATA: IMPACT ON 2D MEASURES

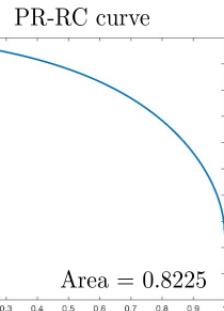
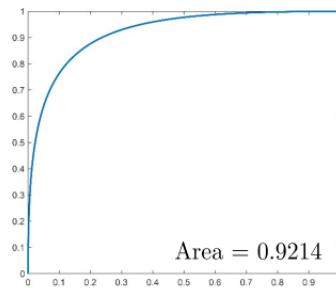
- class-conditional distributions are unit-variance Gaussians with  $\Delta\mu = 2$ ; predictor based on a threshold.
- imbalanced labeled data vs. balanced labeled data



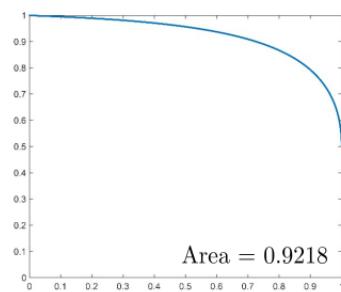
$$P(Y = 1) = \frac{1}{4}$$



$$P(Y = 1) = \frac{1}{2}$$



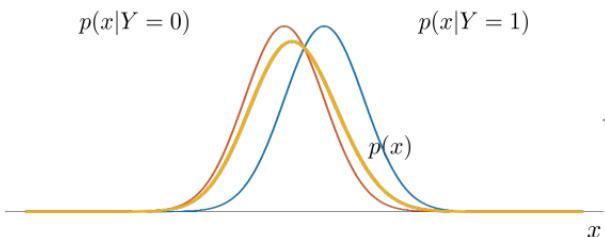
$$\text{Area} = 0.8225$$



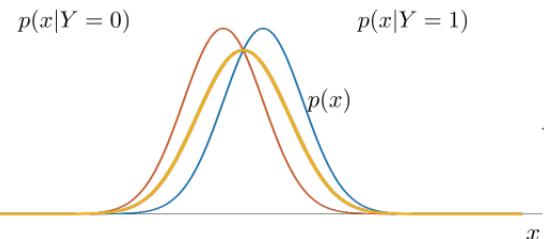
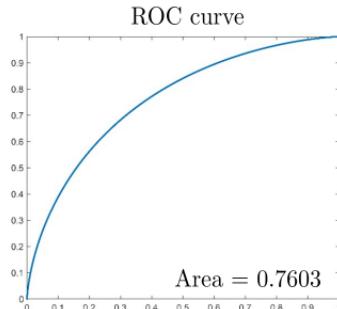
$$\text{Area} = 0.9218$$

# IMBALANCED DATA: IMPACT ON 2D MEASURES

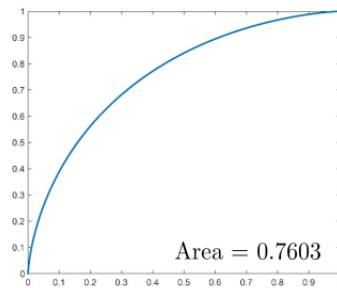
- class-conditional distributions are unit-variance Gaussians with  $\Delta\mu = 1$ ; predictor based on a threshold.
- imbalanced labeled data vs. balanced labeled data



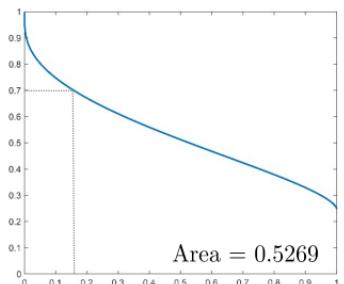
$$P(Y = 1) = \frac{1}{4}$$



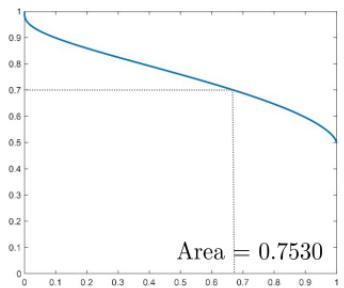
$$P(Y = 1) = \frac{1}{2}$$



PR-RC curve



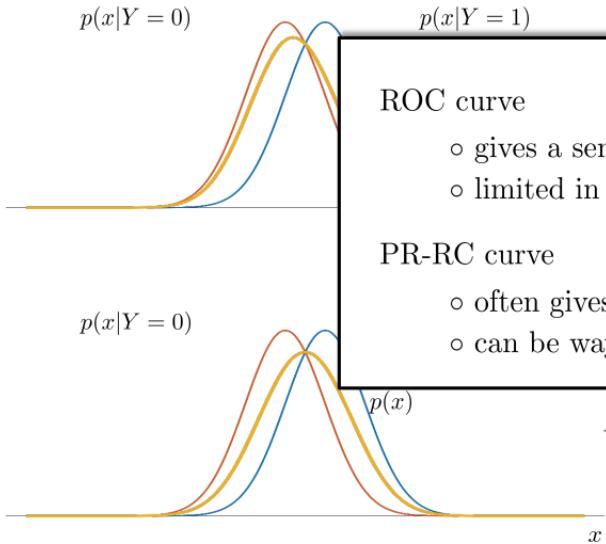
$$\text{Area} = 0.5269$$



$$\text{Area} = 0.7530$$

# IMBALANCED DATA: IMPACT ON 2D MEASURES

- class-conditional distributions are unit-variance Gaussians with  $\Delta\mu = 1$ ; predictor based on a threshold.
- imbalanced labeled data vs. balanced labeled data

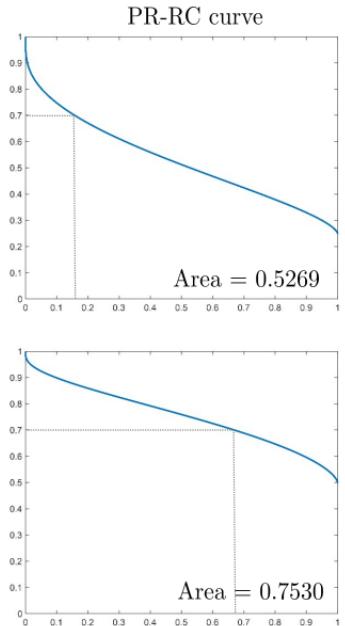
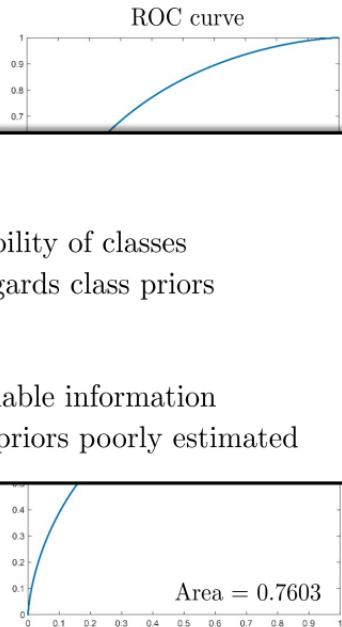


ROC curve

- gives a sense of separability of classes
- limited in that it disregards class priors

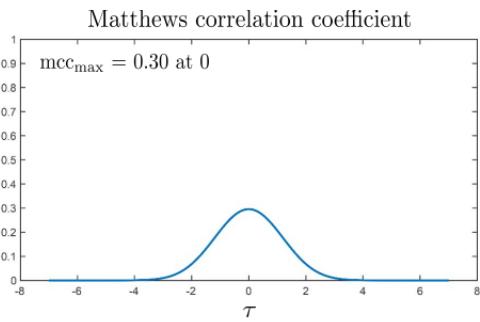
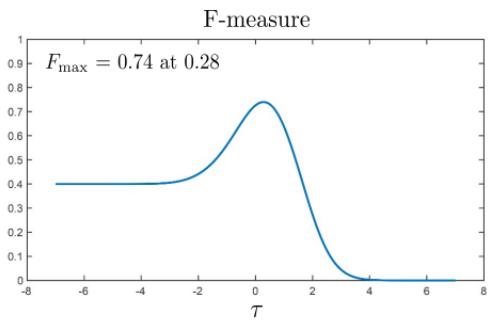
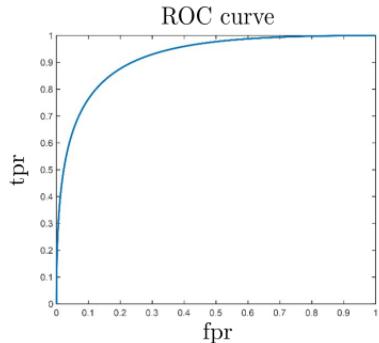
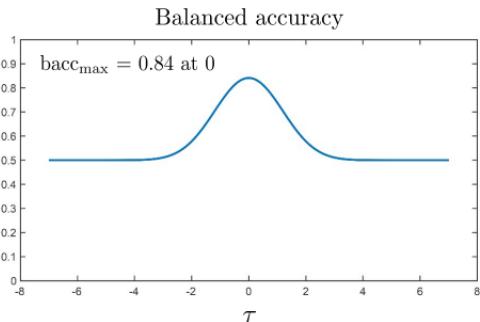
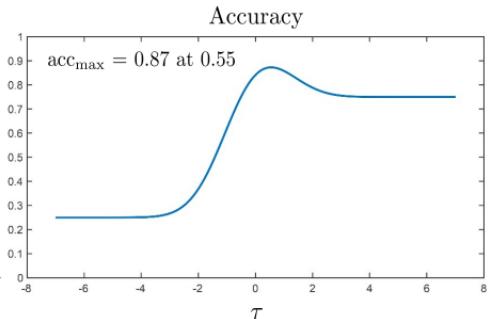
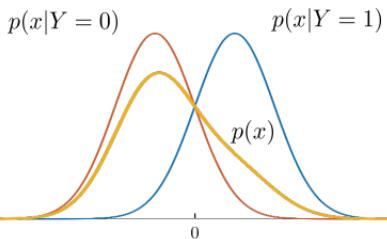
PR-RC curve

- often gives more actionable information
- can be way off if class priors poorly estimated



# IMBALANCED DATA: IMPACT ON 1D MEASURES

- consider  $\Delta\mu = 2$ , symmetric around 0, and  $P(Y = 1) = \frac{1}{4}$



## NOISY CLASS LABELS: MODEL

Before we look at impact on performance estimation, let's look at model

- $Y$  = true but unknown class label
- $Z$  = noisy class label
- $p(x)$  = unchanged by class label noise
- $p(x|z) \neq p(x|y)$  and  $p(z) \neq p(y)$

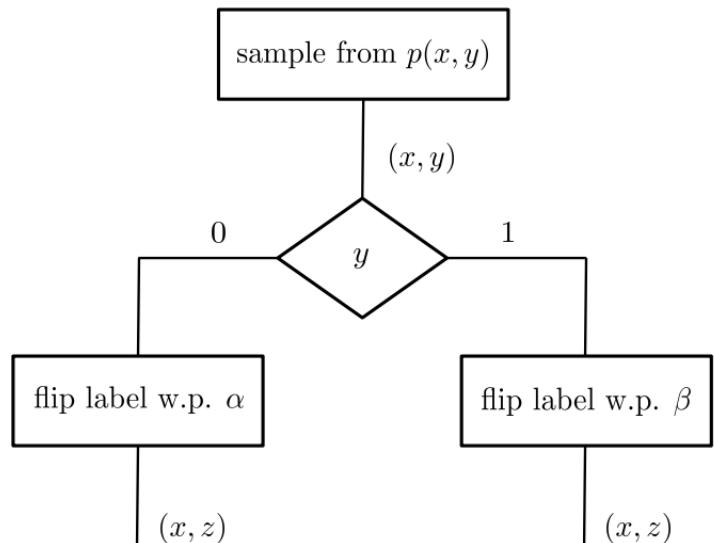
**Expressing  $p(z)$ ,  $p(z|x)$ ,  $p(x|z)$**

$$p_Z(1) = p_Y(1) \cdot (1 - \beta) + p_Y(0) \cdot \alpha$$

$$p(Z = 1|x) = p(Y = 1|x) \cdot (1 - \beta) + p(Y = 0|x) \cdot \alpha$$



flipping labels independent of  $x$



# NOISY CLASS LABELS: MODEL

Class-conditional distributions:

$$p(x|Z=1) = p(x|Y=1) \cdot (1 - \beta') + p(x|Y=0) \cdot \beta'$$

$$p(x|Z=0) = p(x|Y=0) \cdot (1 - \alpha') + p(x|Y=1) \cdot \alpha'$$

where

$$1 - \beta' = \frac{p_Y(1)}{p_Z(1)}(1 - \beta)$$

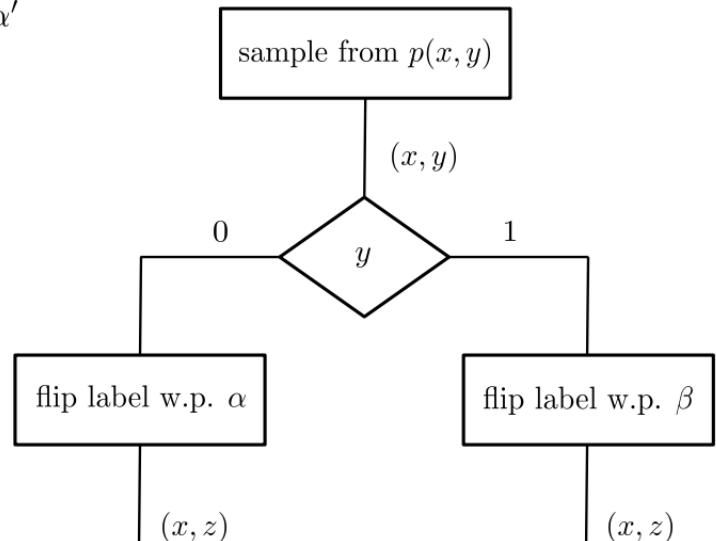
$$1 - \alpha' = \frac{p_Y(0)}{p_Z(0)}(1 - \alpha)$$

**Example:**  $\alpha = 0.05$ ,  $\beta = 0.15$ ,  $p_Y(1) = \frac{1}{4}$

$$p_Z(1) = \frac{1}{4}(1 - \beta) + \frac{3}{4}\alpha = \frac{1}{4}$$

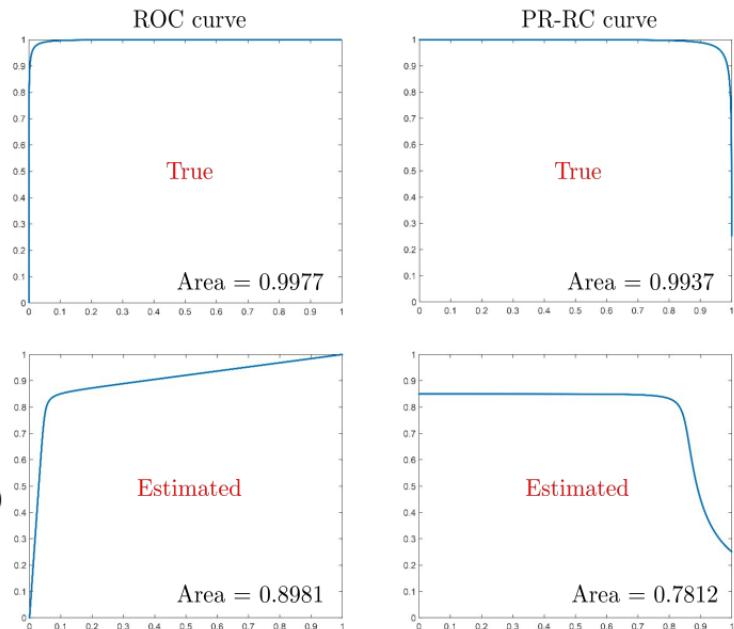
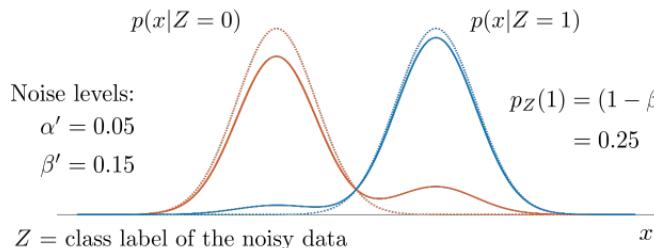
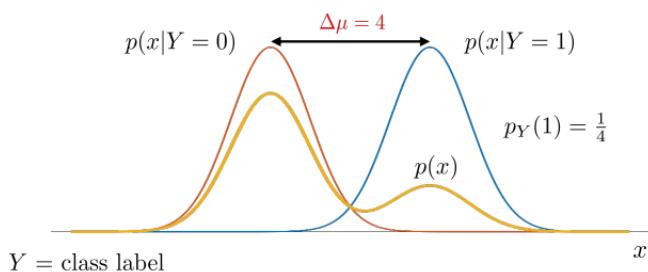
$$\alpha' = \alpha, \beta' = \beta$$

This was a coincidence, because of choices for  $\alpha, \beta, p_Y(1)$



# NOISY DATA: IMPACT ON 2D MEASURES

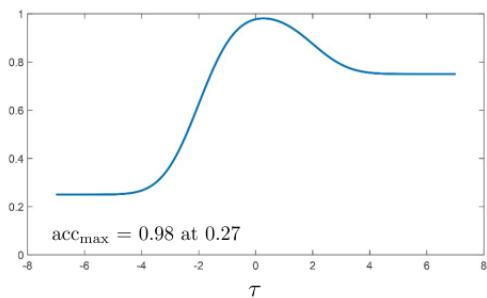
- class-conditional distributions are unit-variance Gaussians with  $\Delta\mu = 4$ ; predictor based on a threshold.
- clean labeled data vs. noisy labeled data



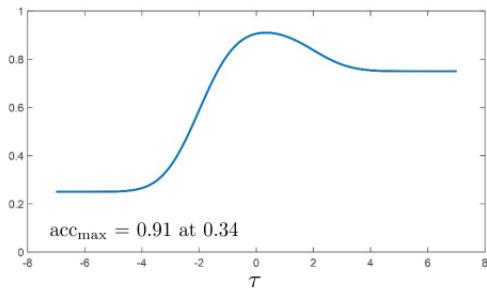
# NOISY DATA: IMPACT ON 1D MEASURES

- class-conditional Gaussians,  $\Delta\mu = 4$ ,  $p_Y(1) = \frac{1}{4}$ ,  $\alpha = 0.05$ ,  $\beta = 0.15$

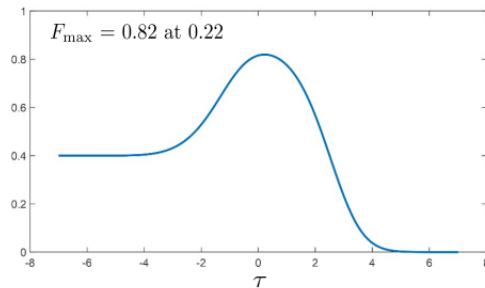
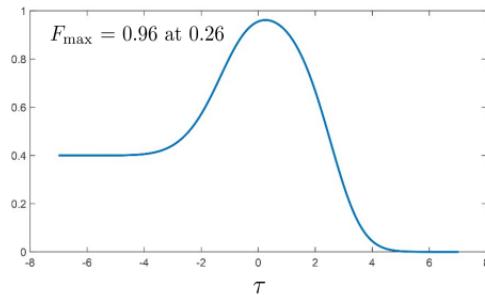
no noise  
(true perf.)



noise  
(measured perf.)

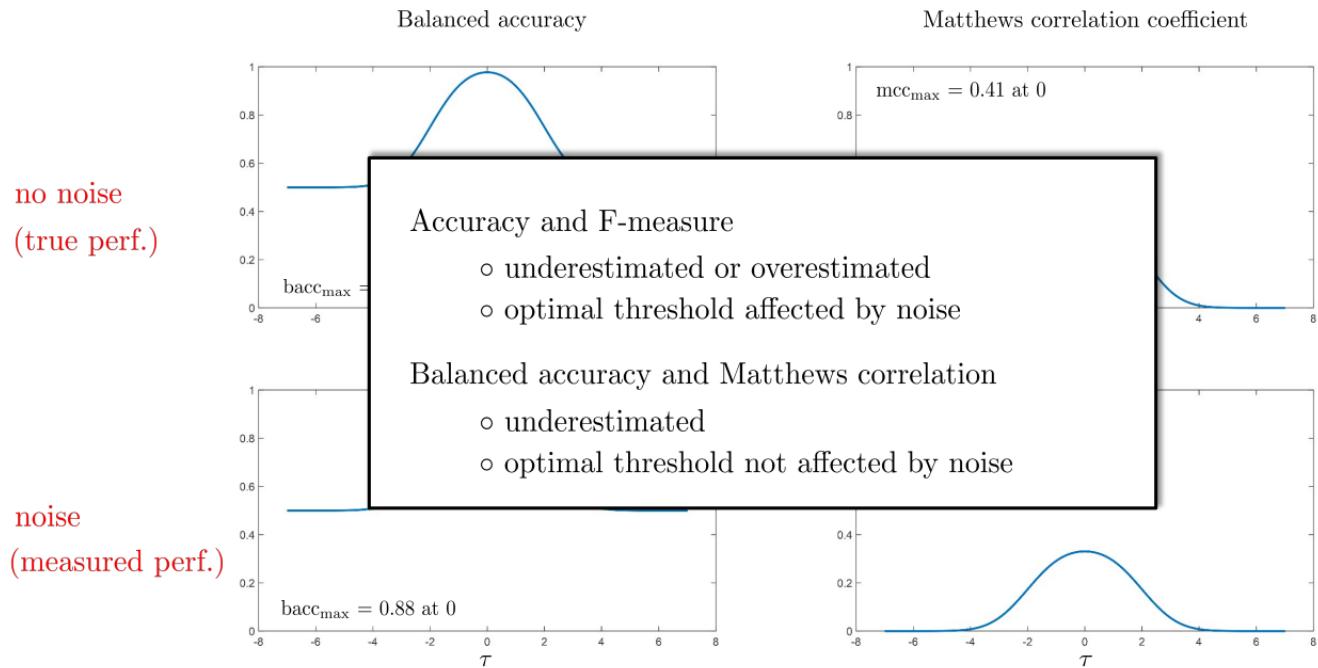


F-measure



# NOISY DATA: IMPACT ON 1D MEASURES

- class-conditional Gaussians,  $\Delta\mu = 4$ ,  $p_Y(1) = \frac{1}{4}$ ,  $\alpha = 0.05$ ,  $\beta = 0.15$



# MULTI-CLASS CLASSIFICATION

**Multidimensional output:**

- $(f(\mathbf{X}), \mathbf{Y})$  = pair of  $m$ -dimensional predictions and outputs
- $\forall j \in \{1, 2, \dots, m\}$  and  $\tau \in \mathbb{R}$  there is a confusion matrix  $C_{j,\tau}$

		Predicted class	
		positive	negative
True class	positive	tp	fn
	negative	fp	tn

Binary classification:  
accuracy =  $\frac{tp+tn}{n}$

**Macro averaging:**

$$\text{accuracy}_M = \frac{1}{m} \sum_{j=1}^m \text{accuracy}_j$$

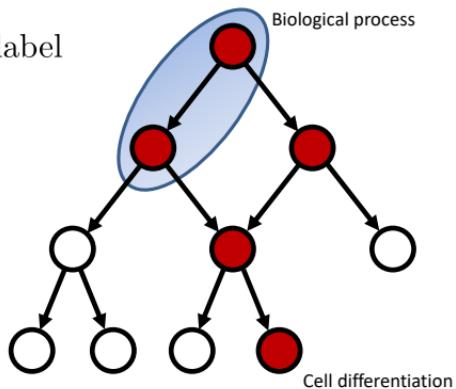
**Micro averaging:**

$$\text{accuracy}_\mu = \frac{1}{mn} \left( \sum_{j=1}^m tp_j + \sum_{j=1}^m tn_j \right)$$

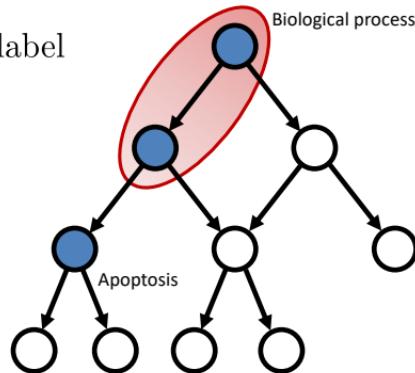
# EVALUATION OF STRUCTURED-OUTPUT PREDICTION

Consistent sub-graph prediction:

Predicted label



True label



Precision:

$$pr = \frac{\#\text{true positives}}{\#\text{positive predictions}} = \frac{2}{5}$$

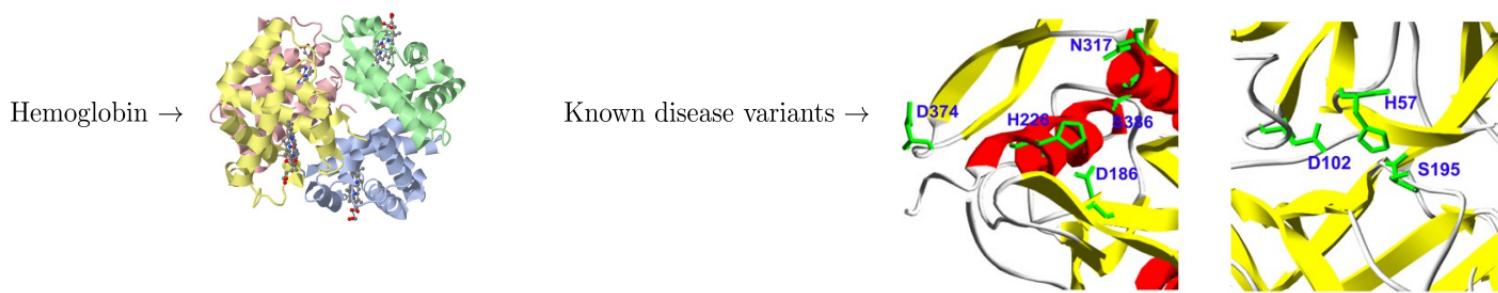
Recall:

$$rc = \frac{\#\text{true positives}}{\#\text{positive terms}} = \frac{2}{3}$$

# PREDICTING IMPACT OF DISEASE VARIANTS

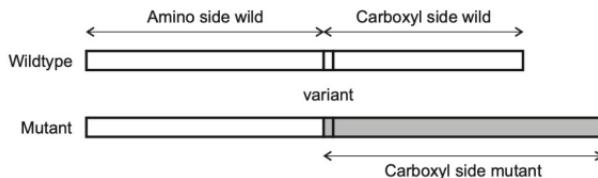
**Problem:** develop/evaluate a pathogenicity predictor of human loss-of-function (LOF) protein variants

**Background:** Proteins are important molecules that do things in our bodies



**LOF variants:** disrupt chunks of proteins

Average healthy human has 50+



# PREDICTING IMPACT OF DISEASE VARIANTS

## Data:

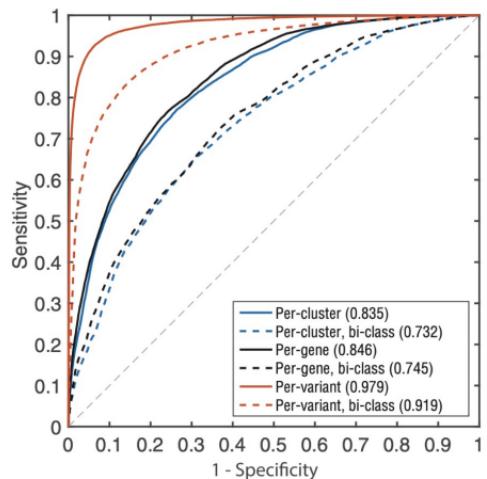
**Table 1.** Number of variants (proteins) present in each data set

	Disease	Neutral	Total
Frameshift	18 116 (1545)	90 135 (13 427)	108 251 (13 713)
Stop gain	14 318 (1681)	7960 (4990)	22 278 (6137)
Total	32 434 (1995)	98 095 (13 605)	

## Issues:

- there are 20,000 genes in the human genome
- these genes are related (gene families)
- some genes do not have known disease mutations
- genes have different lengths and numbers of variants

## How should we evaluate the performance?



## STATISTICAL APPROACH

**Given:** Test data:  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ ,  $x \in \mathcal{X}$  and  $y \in \mathcal{Y}$ , drawn i.i.d. from  $p(x, y)$

Trained model:  $f(x) \in \mathcal{Y}$

$e_T$  = true error of  $f(x)$

$e_S$  = sample error of  $f(x)$  on  $\mathcal{D}$

$X$  = number of errors in a data set of size  $n$

$X_i$  = number of errors on the  $i$ -th example

$$P(X = r) = \binom{n}{r} e_T^r (1 - e_T)^{n-r}$$

$$\mathbb{E}[X] = n e_T$$

$$\mathbb{V}[X] = n e_T (1 - e_T)$$

$e_S$  can be thought of as random variable

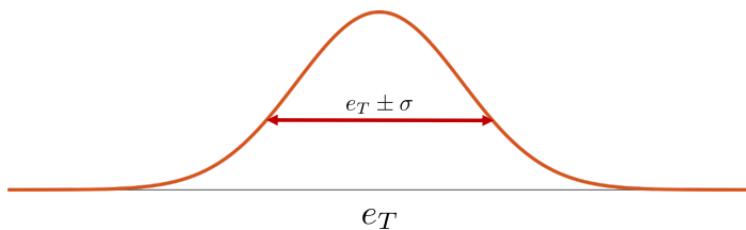
$$\mathbb{E}[e_S] = \mathbb{E} \left[ \frac{1}{n} \sum_{i=1}^n X_i \right] = \frac{1}{n} \mathbb{E} [\sum_{i=1}^n X_i] = \frac{n e_T}{n} = e_T$$

$$\mathbb{V}[e_S] = \mathbb{V} \left[ \frac{1}{n} \sum_{i=1}^n X_i \right] = \frac{1}{n^2} n e_T (1 - e_T) = \frac{e_T (1 - e_T)}{n}$$
$$\frac{e_T (1 - e_T)}{n} \approx \frac{e_S (1 - e_S)}{n}$$

## STATISTICAL APPROACH

Let's approximate binomial distribution by a Gaussian

When  $n e_T (1 - e_T) \geq 5$  we have  $X \sim \mathcal{N}(n e_T, n e_T (1 - e_T))$



If  $e_T$  is known,  $e_S$  is within  $e_T \pm z\sigma$ . This also means, that

$$e_T = e_S \pm z \cdot \sqrt{\frac{e_S(1 - e_S)}{n}}$$

$z = 1$ : 68% confidence interval;  $z = 1.96$ : 95% confidence interval

# COMPARING Two MODELS

Given:

- two models,  $f_1$  and  $f_2$ , evaluated on two different samples,  $\mathcal{S}_1$  and  $\mathcal{S}_2$
- available:  $e_{S_1}$  and  $e_{S_2}$ , say  $e_{S_1} < e_{S_2}$
- not available:  $e_{T_1}$  and  $e_{T_2}$

$$d = e_{T_2} - e_{T_1}$$
$$\hat{d} = e_{S_2} - e_{S_1}$$
$$\sigma_d^2 \approx \frac{e_{S_1}(1-e_{S_1})}{n_1} + \frac{e_{S_2}(1-e_{S_2})}{n_2}$$

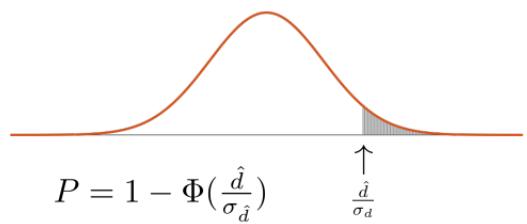
Hypothesis testing (one sided):

$H_0: d = 0$  (the algorithms are equal)

$H_1: d > 0$  ( $f_1$  is better than  $f_2$ )

$$\frac{\hat{d}-d}{\sigma_d} : \mathcal{N}(0, 1) \rightarrow$$

P-value: area under the tail



$$P = 1 - \Phi\left(\frac{\hat{d}}{\sigma_{\hat{d}}}\right)$$

# BOOTSTRAPPING

- we have an estimator  $\hat{\theta}$  of parameter  $\theta$
- a general nonparametric procedure for evaluating dispersion of  $\hat{\theta}$

**Given:** data set  $\mathcal{D} = \{x_i\}_{i=1}^n$ ,  $x \in \mathcal{X}$  and some estimator  $\hat{\theta} = g(\mathcal{D})$

## Procedure:

For  $b = 1$  to  $B$

- draw sample  $\mathcal{D}_b^*$  w/ replacement from  $\mathcal{D}$
- compute  $g(\mathcal{D}_b^*)$

end

$$g^*(\mathcal{D}) = \frac{1}{B} \sum_{b=1}^B g(\mathcal{D}_b^*)$$

$$\hat{\sigma}^2 = \frac{1}{B-1} \sum_{b=1}^B (g(\mathcal{D}_b^*) - g^*(\mathcal{D}))^2$$

## Examples:

$g(\mathcal{D})$  = sample mean

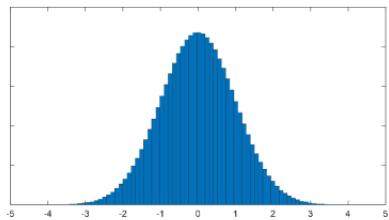
$g(\mathcal{D})$  = AUC when  $\mathcal{D}$  are prediction scores and true classes

← standard error (SE) estimated as  $\sqrt{\hat{\sigma}^2}$

# BOOTSTRAPPING

- standard error: true  $\theta$  is within  $\hat{\theta} \pm \text{SE}$  w.p. 68%

$$X \sim \mathcal{N}(0, 1)$$



← normal distribution

- standard error =  $\frac{\hat{\sigma}}{\sqrt{n}}$

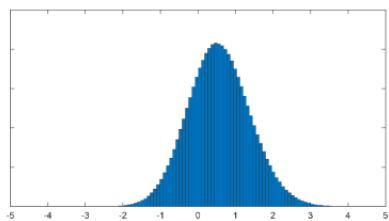
**Example:**

$$n = 100$$

$$\mu = \hat{\mu} \pm 1.96 \cdot \text{SE}$$

↑  
95% c.i.

$$X \sim \max \{\mathcal{N}(0, 1), \mathcal{N}(0, 1)\}$$



← skew normal distribution

- asymmetric  $100 \cdot \alpha$  confidence intervals determined by thresholds separating  $\frac{1-\alpha}{2}$  smallest and  $\frac{1-\alpha}{2}$  largest scores

**Example:**

$$n = 100$$

$$\mu \in (-1.30, 2.00)$$

↑  
95% c.i.

# COMPARING LEARNING ALGORITHMS

**Model:**  $f : \mathcal{X} \rightarrow \mathcal{Y}$

**Algorithm:**  $a : (\mathcal{X} \times \mathcal{Y})^n \rightarrow \mathcal{F}$

	$a_1$	$a_2$
$\mathcal{D}_1$	1	0
$\mathcal{D}_2$	1	0
$\vdots$		
$\mathcal{D}_K$	0	1

- $K$  datasets
- score wins (1s) vs. losses (0s)
- $a_1$  scored  $k$  wins

**Hypothesis testing (one sided):**

$H_0$ :  $a_1$  and  $a_2$  are equal

$H_1$ :  $a_1$  is better than  $a_2$

$$\rightarrow \quad P = \sum_{i=k}^K \binom{K}{j} \alpha^j (1-\alpha)^{K-j} \quad \alpha = \frac{1}{2}$$

- probability that  $k$  or more wins would be observed by chance (under  $H_0$ )

# CONCLUDING REMARKS

## **Empirical evaluation:**

- Core constitutive part of machine learning
- Science and art

## **Addtional issues:**

- Reliability: extensive auditing to provide uncertainty guarantees that lead to actionable evidence
- Interpretability: models must be able to point out the factors that influence their decisions
- Fairness: models must be fair and must not cause unintended harm to users upon deployment