Introduction to Machine Learning:

Binary Classifier Performance Evaluation

ECE 580
Spring 2022

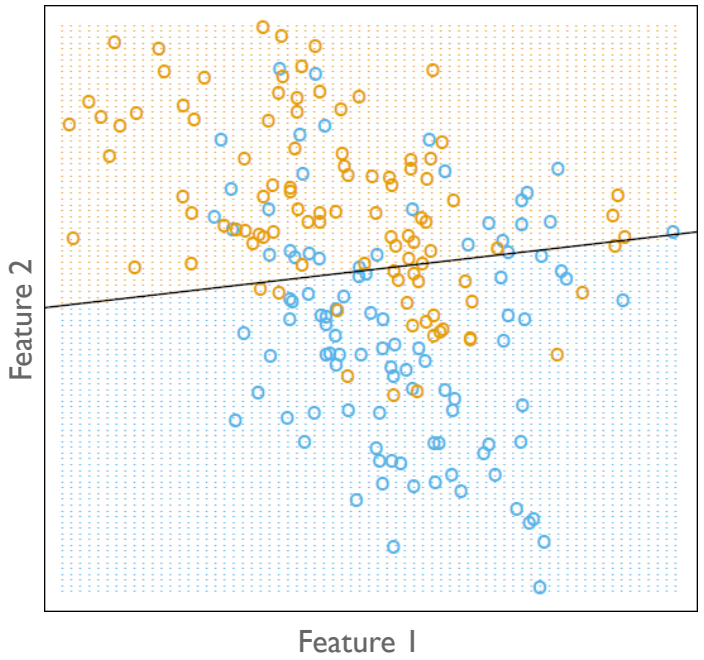
Stacy Tantum, Ph.D.

Why Evaluate Classifier Performance?

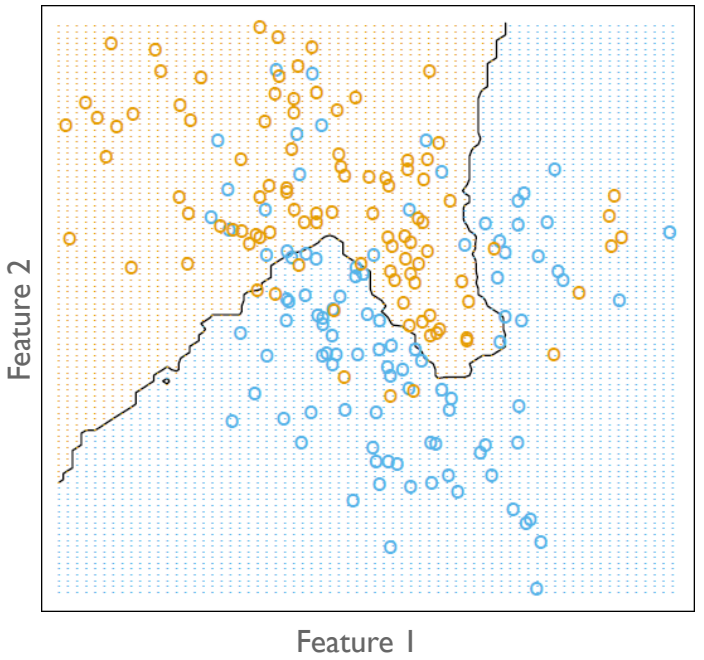
Choose among candidate classifiers

Bias-Variance Trade-Off

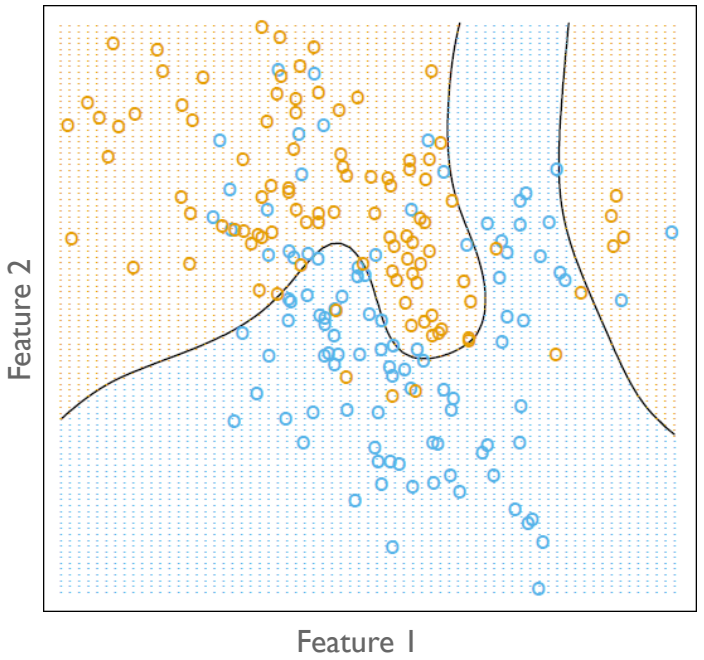
Linear Classifier



Nearest Neighbor Classifier



Bayes Classifier



"No Free Lunch" Theorem

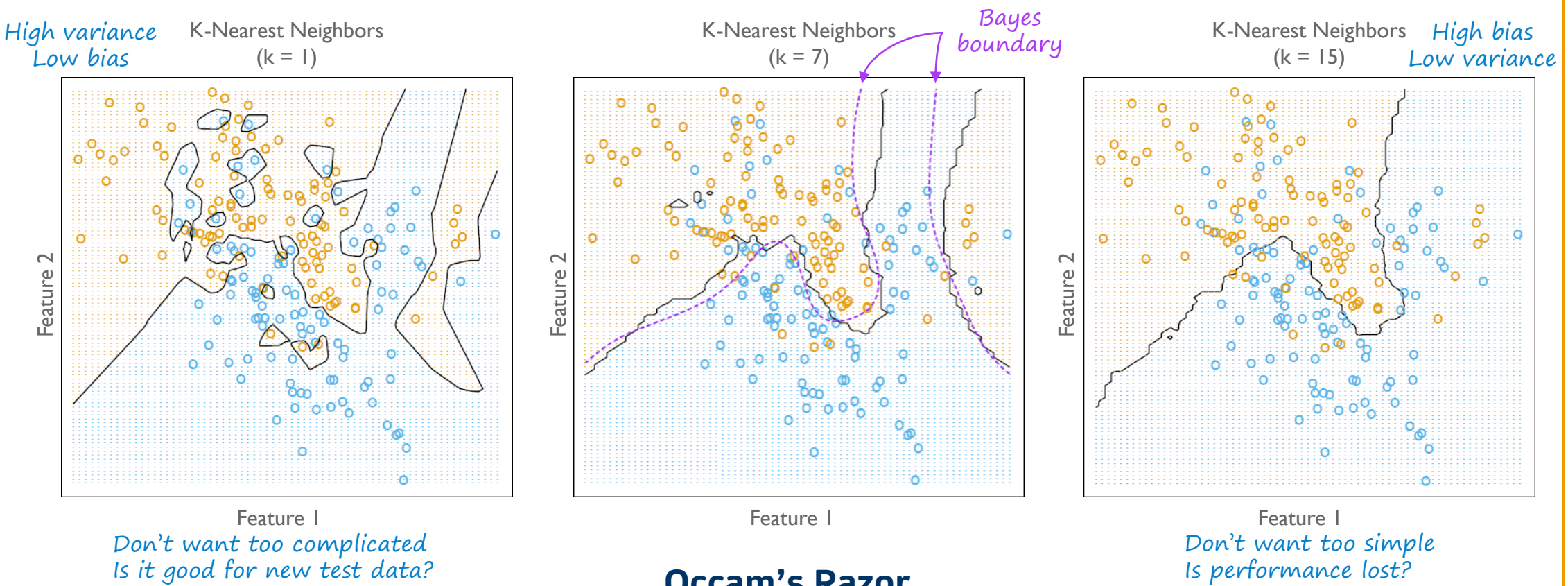
No classifier is inherently superior (or inferior) to all others

According to Wikipedia:
William of Ockham was an English Franciscan friar, scholastic philosopher, and theologian who preferred simplicity in defending the idea of divine miracles – "the simplest solution is most likely the right one"

Why Evaluate Classifier Performance?

Compare/choose classifier parameter(s)

Bias-Variance Trade-Off



Don't want too complicated
Is it good for new test data?

Don't want too simple
Is performance lost?

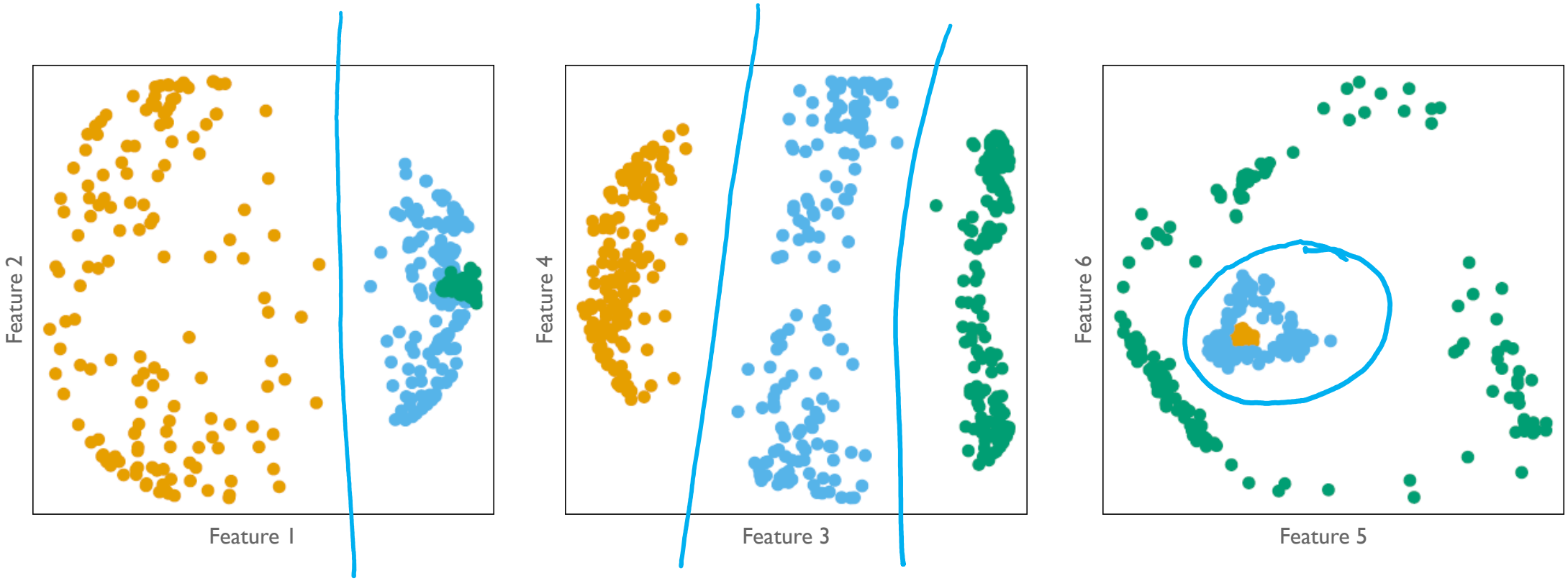
Occam's Razor
(avoiding overfitting/overtraining)

Classifiers should be no more complicated than necessary

Why Evaluate Classifier Performance?

Compare/choose feature subsets

Bias-Variance Trade-Off



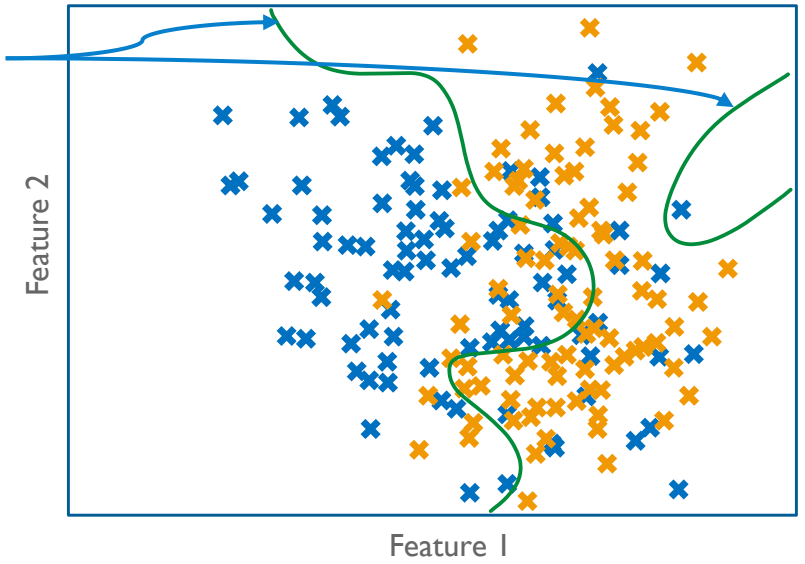
"Ugly Duckling" Theorem

No feature representation is inherently superior (or inferior) to all others

These 3 theorems are why the answer to many questions is "it depends..."

Why Evaluate Classifier Performance?

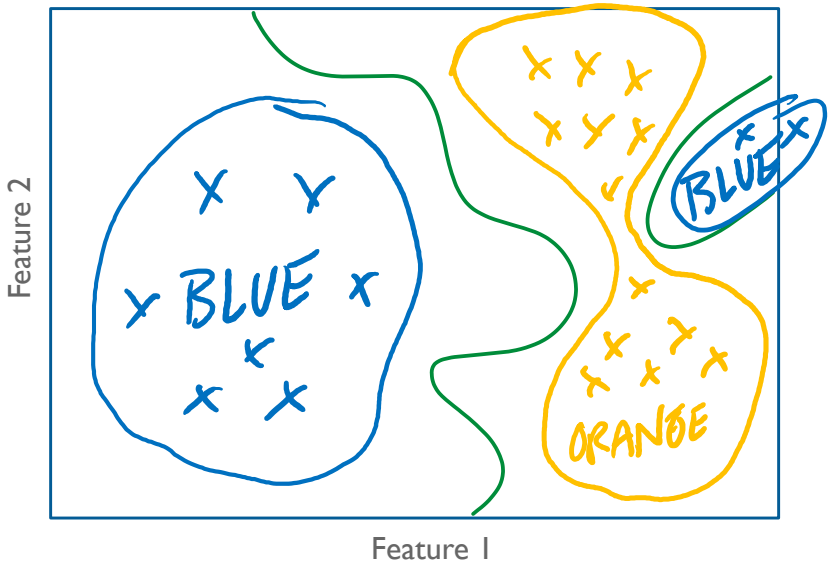
Not much training data here...
where should boundary be?



Predict likely performance when **deployed**

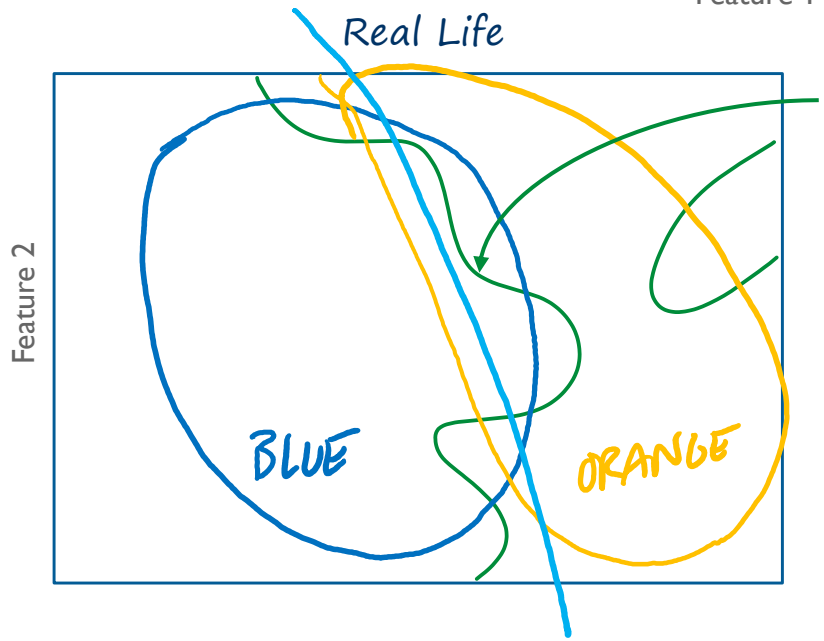
(operating on data not seen during development)

Desired Outcome / Ideal Situation



Real Life

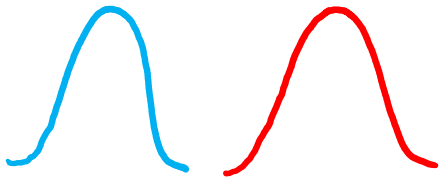
Is this boundary too complicated?

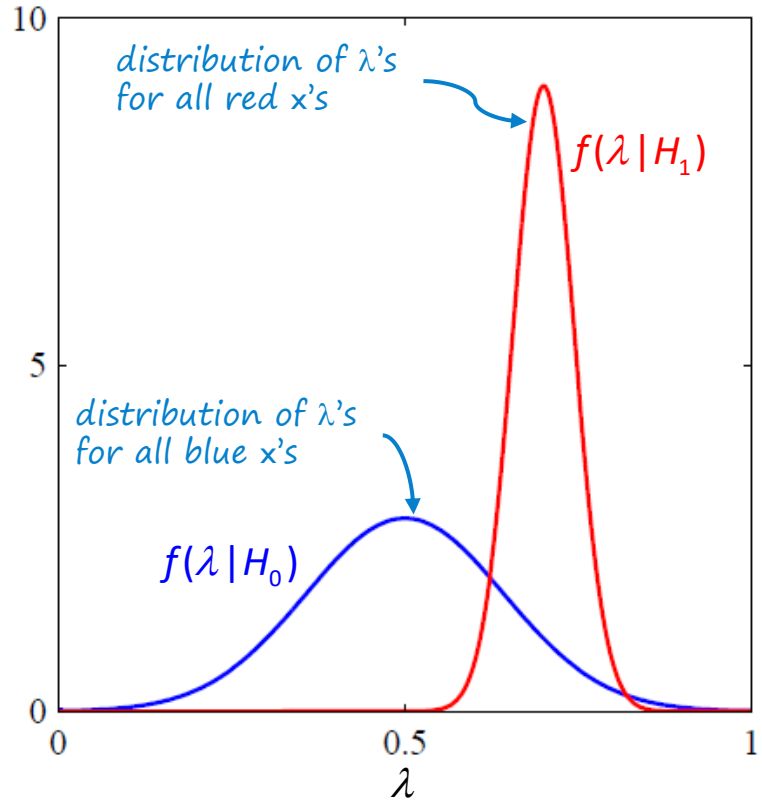


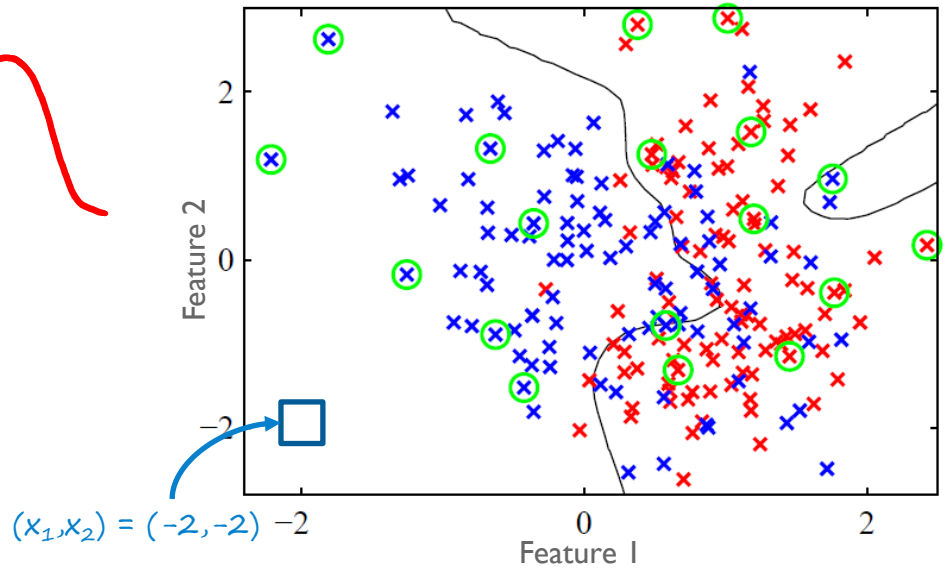
Distributions of Decision Statistics

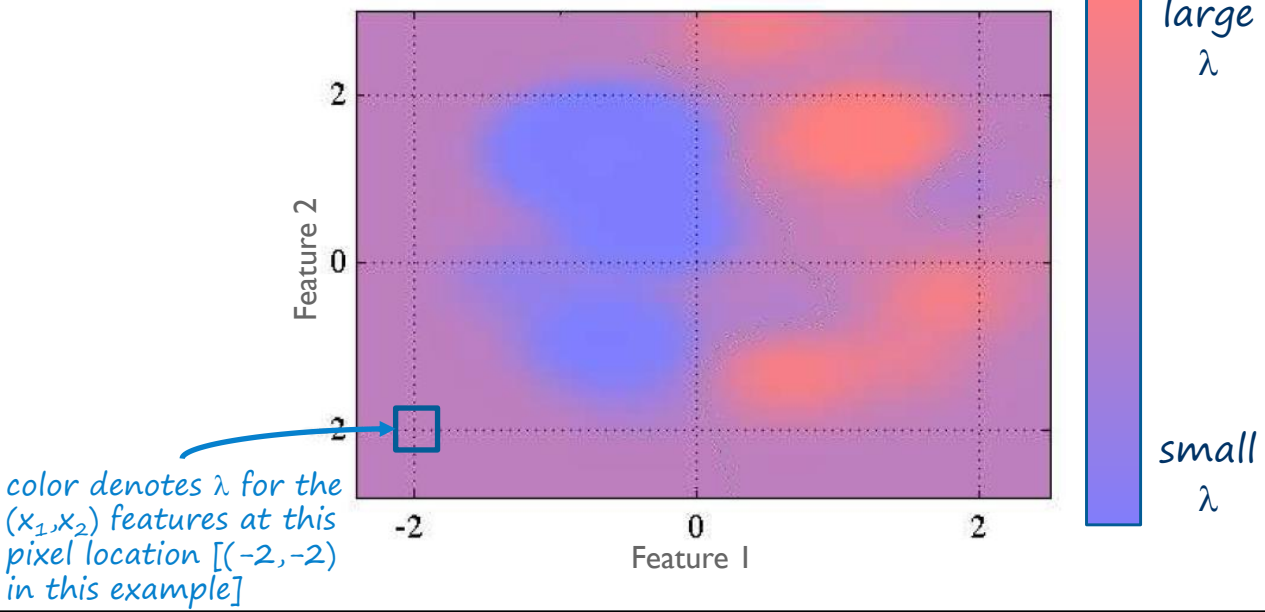
pdfs of decision statistics for:

- All H_0 data, $f(\lambda \mid H_0)$
- All H_1 data, $f(\lambda \mid H_1)$









Binary Decision Performance Evaluation

$P_{CR}(\beta) = p(\text{decide } H_0 | \text{data from } H_0, \beta)$
(want ↑) $= p(\lambda < \beta | \text{data from } H_0)$

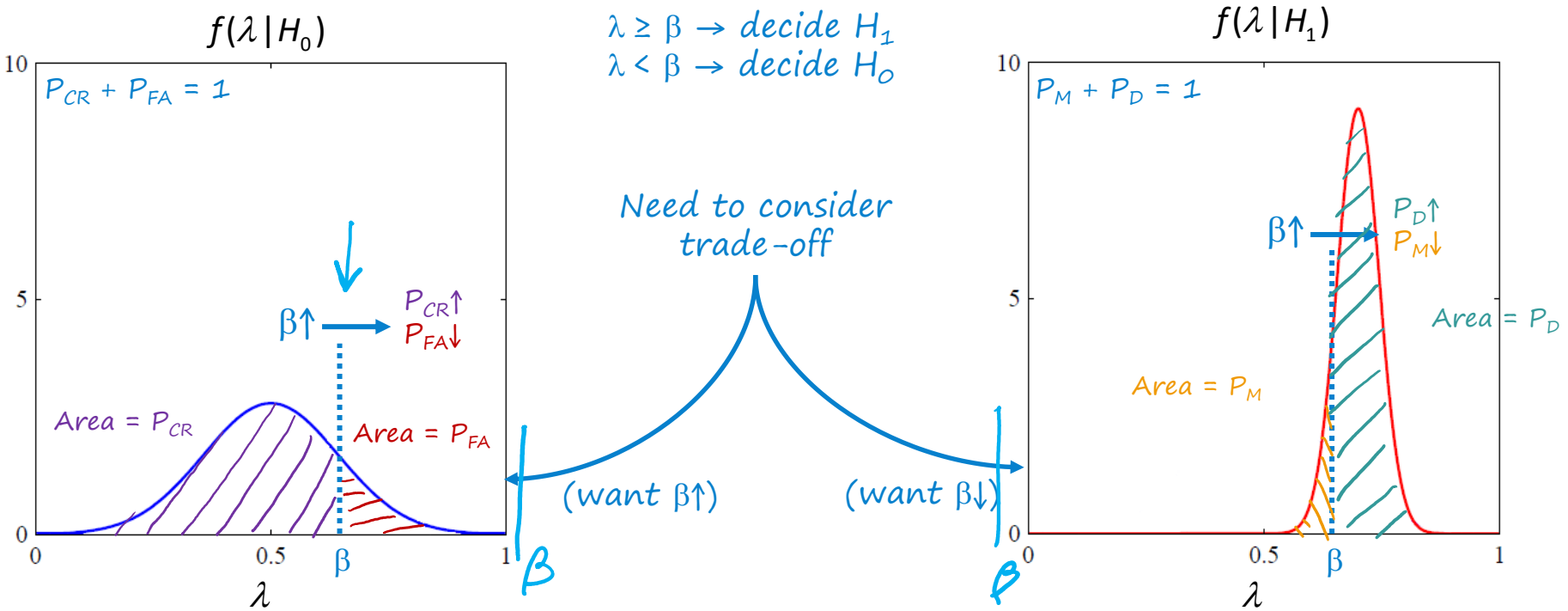
$P_{FA}(\beta) = p(\text{decide } H_1 | \text{data from } H_0, \beta)$
(want ↓) $= p(\lambda \geq \beta | \text{data from } H_0)$

$\beta \uparrow$

$P_M(\beta) = p(\text{decide } H_0 | \text{data from } H_1, \beta)$
(want ↓) $= p(\lambda < \beta | \text{data from } H_1)$

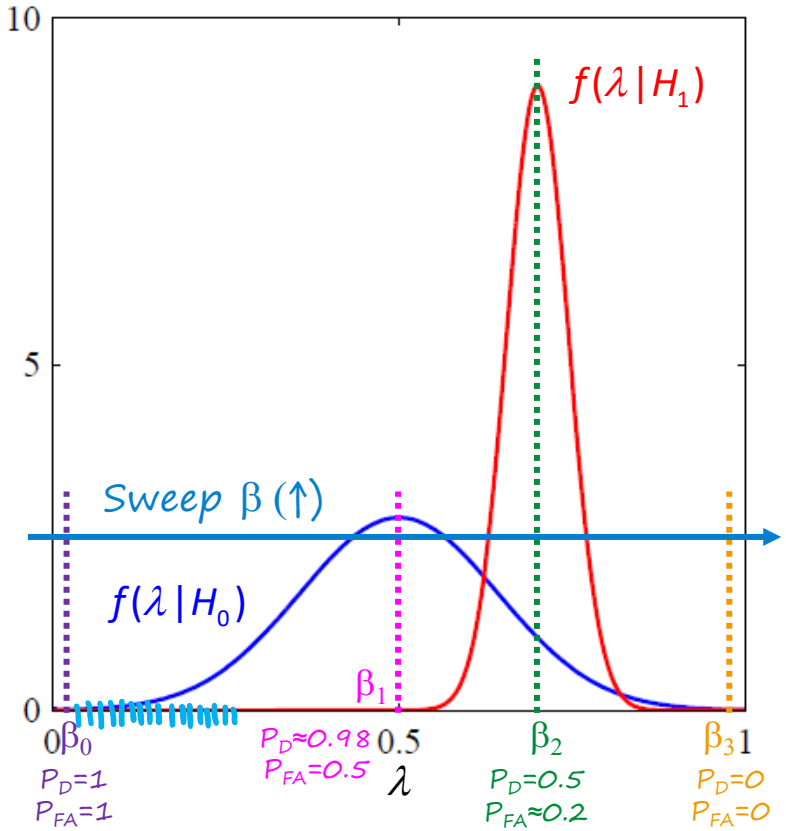
$P_D(\beta) = p(\text{decide } H_1 | \text{data from } H_1, \beta)$
(want ↑) $= p(\lambda \geq \beta | \text{data from } H_1)$

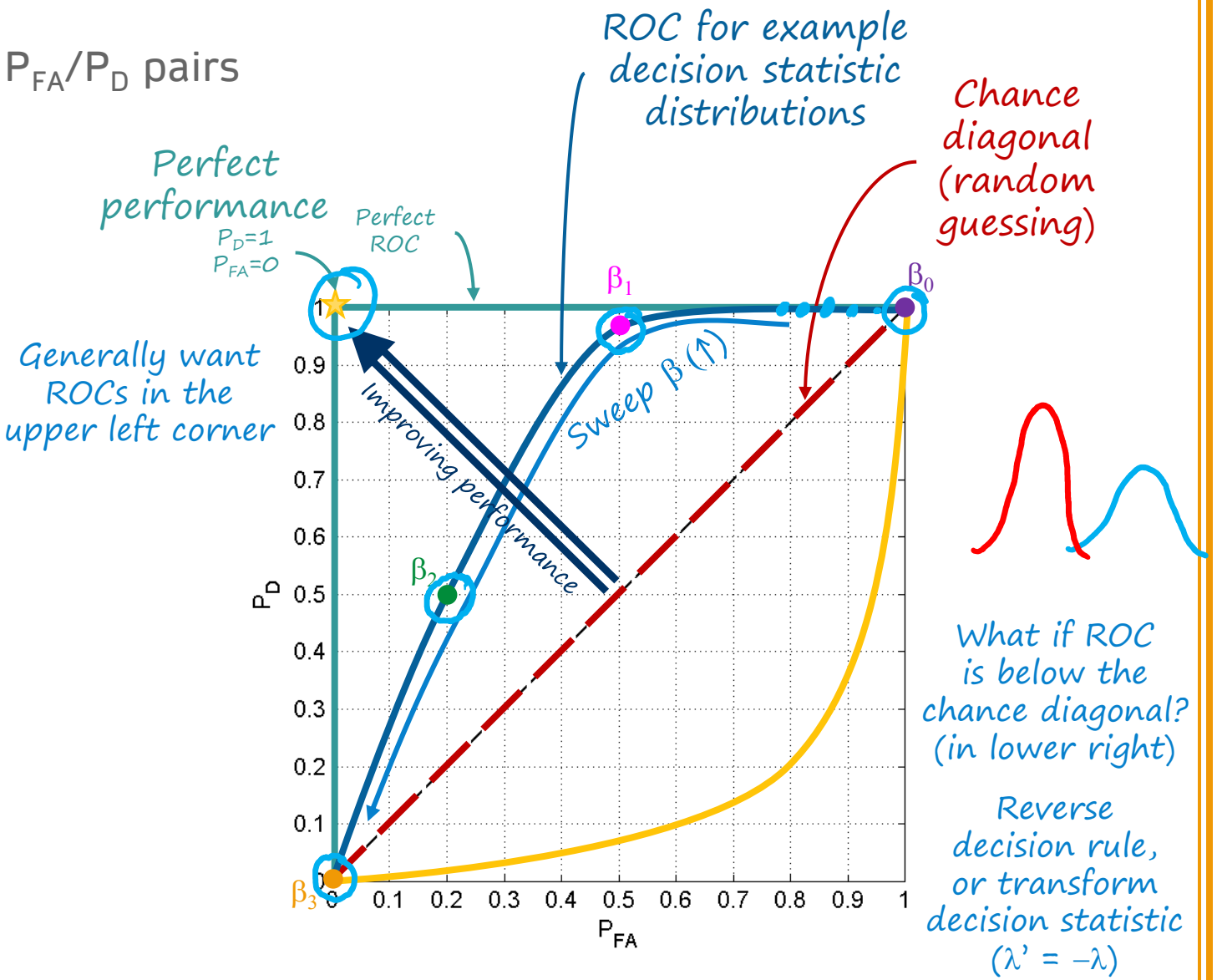
$\beta \downarrow$



Generating a ROC (Receiver Operating Characteristic)

Sweep the threshold to generate P_{FA}/P_D pairs





ROC vs Sensitivity-Specificity

$(P_D \quad - \quad P_{FA})$

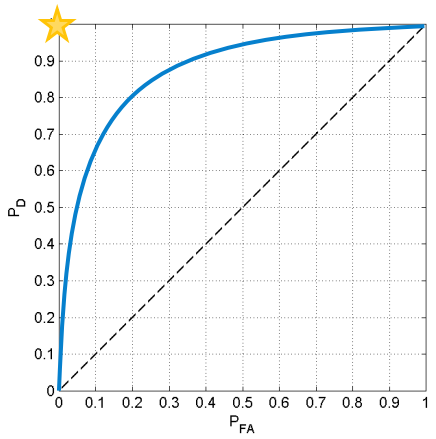
ROC (Sensitivity – Fall-out)

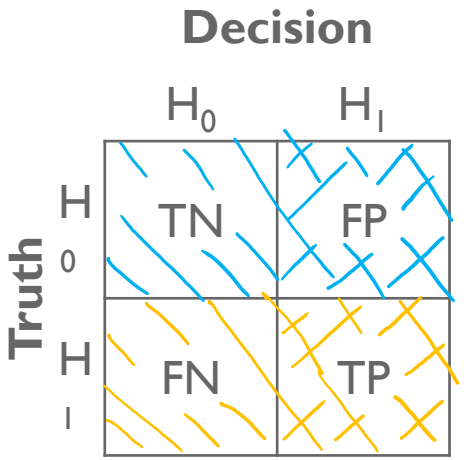
$$\text{Sensitivity} = P_D = \frac{TP}{TP + FN}$$

$$\text{Fall-out} = P_{FA} = \frac{FP}{FP + TN}$$

A change in any of TN, FP, FN, TP changes P_D or P_{FA} (ROC changes)

→ ROC captures ***everything***





$(P_D \quad - \quad P_{CR})$

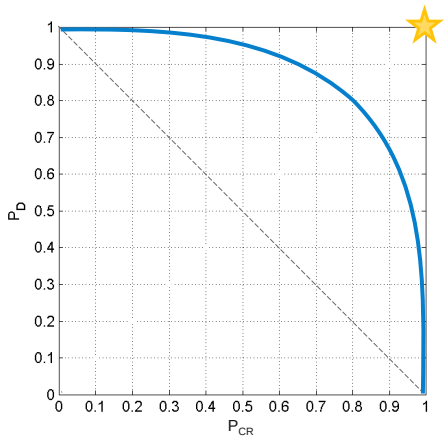
Sensitivity – Specificity

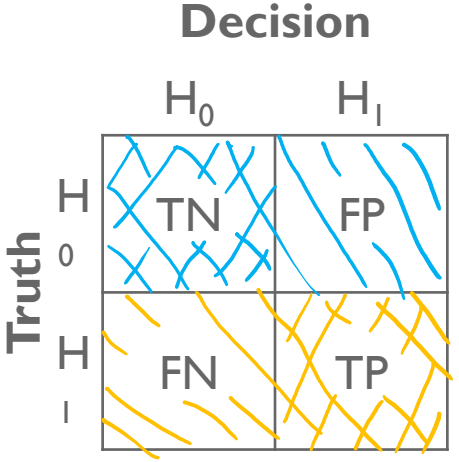
$$\text{Sensitivity} = P_D = \frac{TP}{TP + FN}$$

$$\text{Specificity} = P_{CR} = \frac{TN}{FP + TN} = 1 - P_{FA}$$

A change in any of TN, FP, FN, TP changes P_D or P_{CR} (curve changes)

→ Curve captures ***everything***





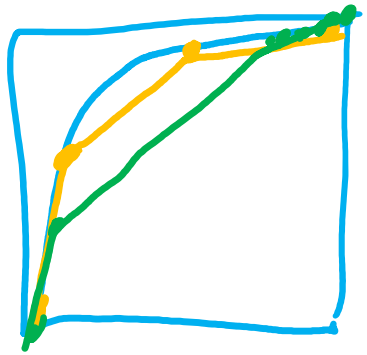
Generating ROCs in Practice

$$P_{CR}(\beta) = \frac{\#H_0 \ \lambda's < \beta}{\#H_0 \ \lambda's}$$

$$P_{M}(\beta) = \frac{\#H_1 \ \lambda's < \beta}{\#H_1 \ \lambda's}$$

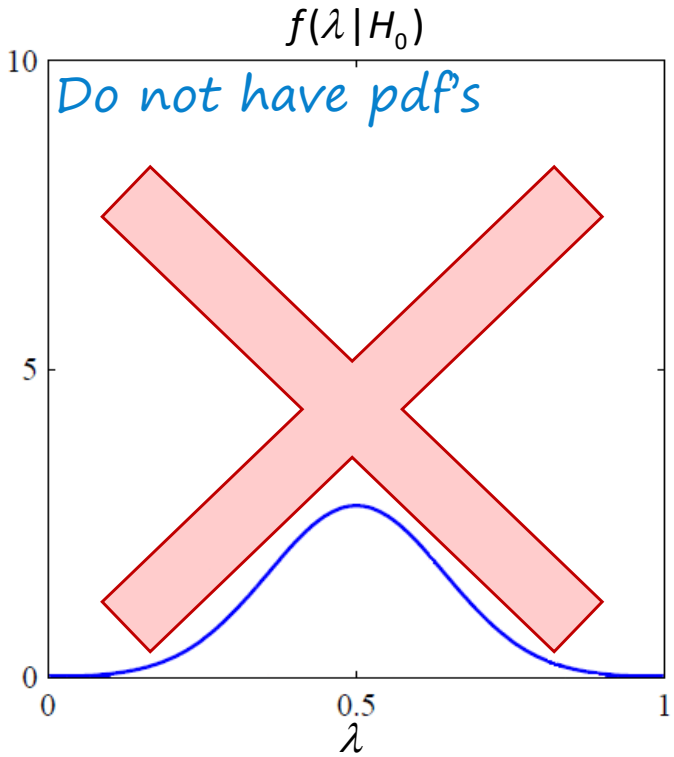
$$P_{FA}(\beta) = \frac{\#H_0 \ \lambda's \geq \beta}{\#H_0 \ \lambda's}$$

$$P_{D}(\beta) = \frac{\#H_1 \ \lambda's \geq \beta}{\#H_1 \ \lambda's}$$

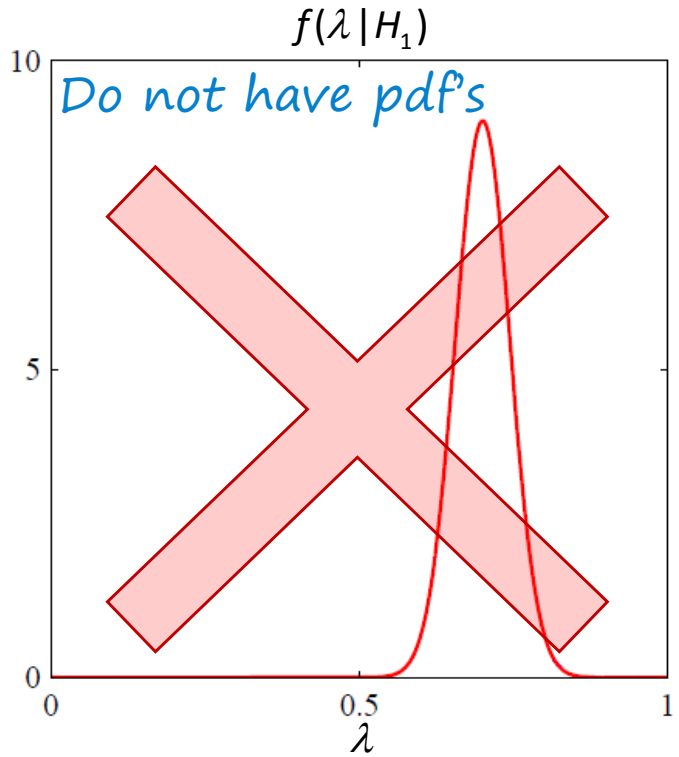


Have a list of λ's

Have a list of λ's



λ	truth
0.56	0
0.35	0
0	0
0.21	0
0.11	0



λ	truth
0.18	1
0.92	1
0.42	1
0.88	1
0.82	1