IAI, TCG CREST
Avisek Gupta (avisek003@gmail.com)

Machine Learning

17 – ROC Analysis, No Free Lunch, PAC Learning

November 15, 2022

Binary Classification: TP, TN, FP, FN

For binary classification $k = 2$, we call a class c_1 the **positive** class, and the other class c_2 as the **negative** class. We obtain a 2×2 confusion matrix, whose entries have the following names.

	R_1 (Predicted Positive)	R_2 (Predicted Negative)
D_1 (GT Positive)	True Positive (TP)	False Negative (FN)
D_2 (GT Negative)	False Positive (FP)	True Negative (TN)

True Positives (TP): The number of positive-class instances that have been classified correctly.

$$\mathrm{TP} = n_{11} = |\{x_i | \hat{y}_i = y_i = c_1\}|$$

True Negatives (TN): The number of negative-class instances that have been classified correctly.

$$\mathrm{TN} = n_{22} = |\{x_i | \hat{y}_i = y_i = c_2\}|$$

Binary Classification: TP, TN, FP, FN

For binary classification $k = 2$, we call a class c_1 the **positive** class, and the other class c_2 as the **negative** class. We obtain a 2×2 confusion matrix, whose entries have the following names.

	R_1 (Predicted Positive)	R_2 (Predicted Negative)
D_1 (GT Positive)	True Positive (TP)	False Negative (FN)
D_2 (GT Negative)	False Positive (FP)	True Negative (TN)

False Positives (FP): The number of instances that have been incorrectly classified as positive.

$$\mathrm{FP} = n_{21} = |\{x_i | \hat{y}_i = c_1 \text{ and } y_i = c_2\}|$$

False Negatives (FN): The number of instances that have been incorrectly classified as negative.

$$\mathrm{FN} = n_{12} = |\{x_i | \hat{y}_i = c_2 \text{ and } y_i = c_1\}|$$

Binary Classification: Accuracy, Precision

	R_1 (Predicted Positive)	R_2 (Predicted Negative)
D_1 (GT Positive)	True Positive (TP)	False Negative (FN)
D_2 (GT Negative)	False Positive (FP)	True Negative (TN)

Accuracy:

$$ACC = \frac{TP + TN}{n}$$

Error Rates:

$$ER = \frac{FP + FN}{n}$$

Binary Classification: Accuracy, Precision

	R_1 (Predicted Positive)	R_2 (Predicted Negative)
D_1 (GT Positive)	True Positive (TP)	False Negative (FN)
D_2 (GT Negative)	False Positive (FP)	True Negative (TN)

Accuracy:

$$ACC = \frac{TP + TN}{n}$$

Error Rates:

$$ER = \frac{FP + FN}{n}$$

Positive-class Precision:

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

Negative-class Precision:

$$Precision_N = \frac{TN}{TN + FN}$$

Binary Classification: TPR, FPR

	R_1 (Predicted Positive)	R_2 (Predicted Negative)
D_1 (GT Positive)	True Positive (TP)	False Negative (FN)
D_2 (GT Negative)	False Positive (FP)	True Negative (TN)

True Positive Rate (Sensitivity):

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

True Negative Rate (Specificity):

$$TNR = Recall_N = \frac{TN}{TN + FP}$$

False Positive Rate:

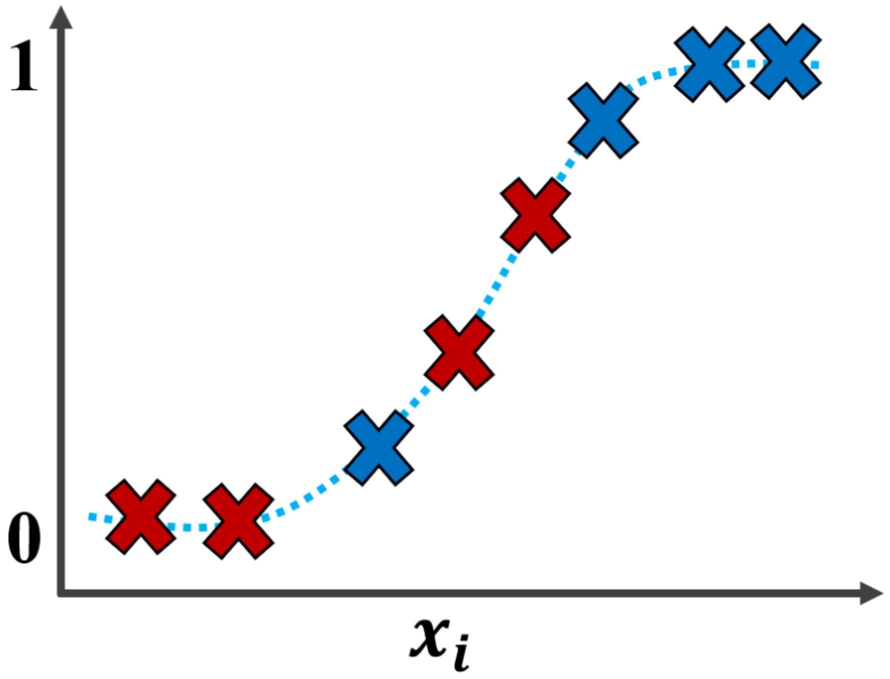
$$FPR = \frac{FP}{FP + TN} = 1 - Recall_N$$

False Negative Rate:

$$FNR = \frac{FN}{FN + TP} = 1 - Recall_P$$

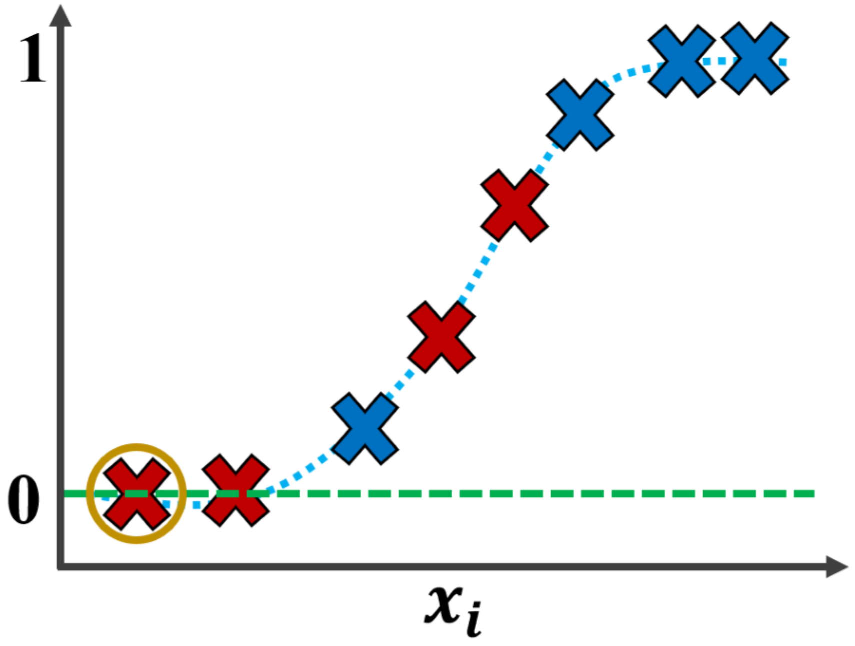
Receiver Operating Characteristics (ROC) Analysis

- For binary classification, ROC analysis can help to (i) identify optimal parameter settings for a classifier (ii) compare two classifiers.
- ROC analysis requires a classifier to output a **score** for each instances $S(\mathbf{x}_i)$. E.g., in Logistic Regression, the score can be the distance of an instance to the hyperplane.



Receiver Operating Characteristics (ROC) Analysis

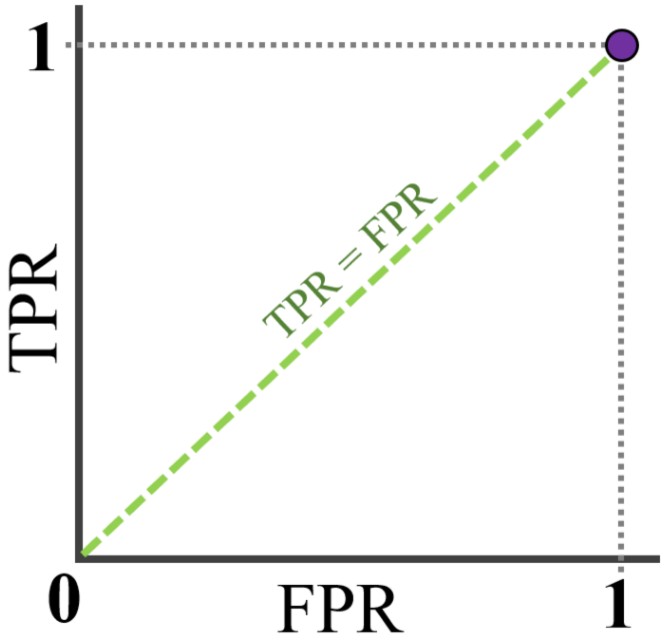
- For a threshold ρ, scores above ρ are classified to the positive class, the rest are classified to the negative class.
- For a range of possible values of ρ, the TPR (y-axis) vs the FPR (x-axis) are tracked. The resulting plot is the ROC curve.



	Pred P	Pred N
GT P	4	0
GT N	4	0

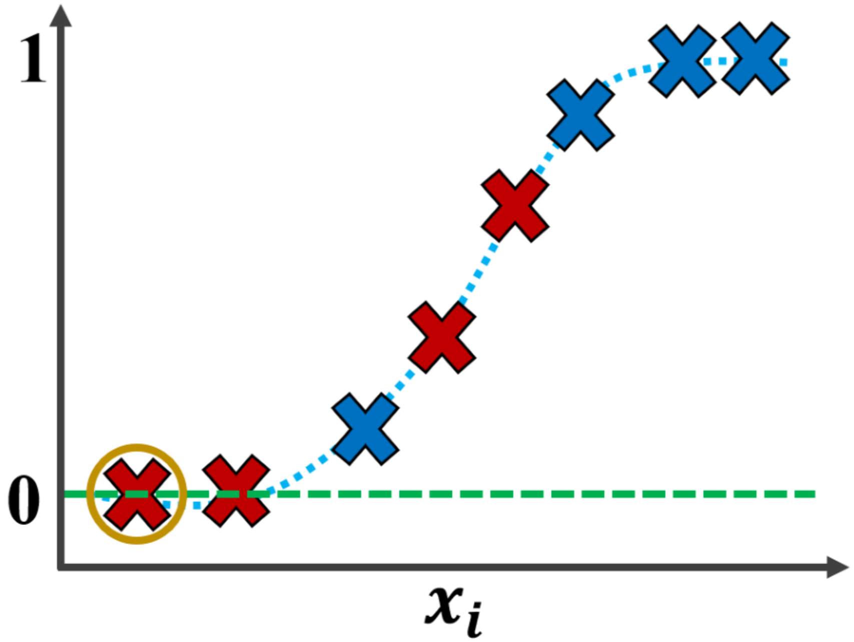
$$TPR = \frac{4}{4+0} = 1$$

$$FPR = \frac{4}{4+0} = 1$$



Example from: Statquest with Josh Stramer, https://www.youtube.com/watch?v=4jRBRDbJemM

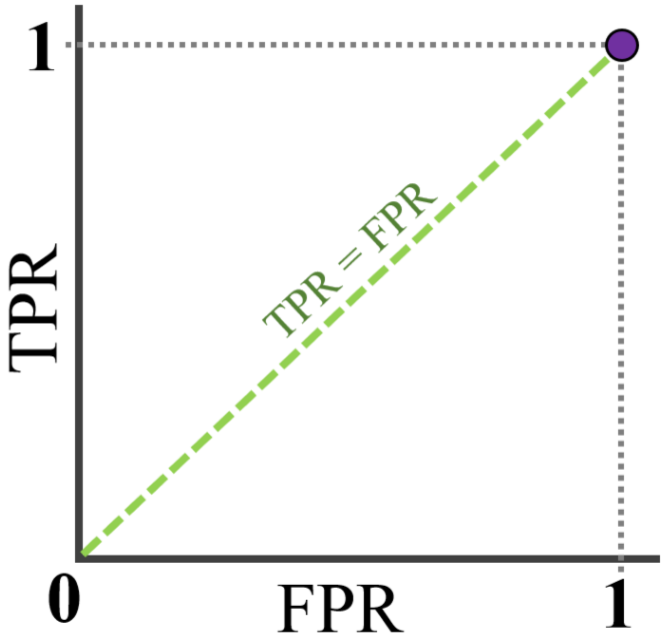
Receiver Operating Characteristics (ROC) Analysis



	Pred P	Pred N
GT P	4	0
GT N	4	0

$$TPR = \frac{4}{4+0} = 1$$

$$FPR = \frac{4}{4+0} = 1$$



We consider a minimum and maximum possible values for ρ:

$$\rho^{\min} = \min_i \{S(\mathbf{x}_i)\}, \quad \rho^{\max} = \max_i \{S(\mathbf{x}_i)\}$$

For distinct values of ρ in the range of $[\rho^{\min}, \rho^{\max}]$, the set of positive points are:

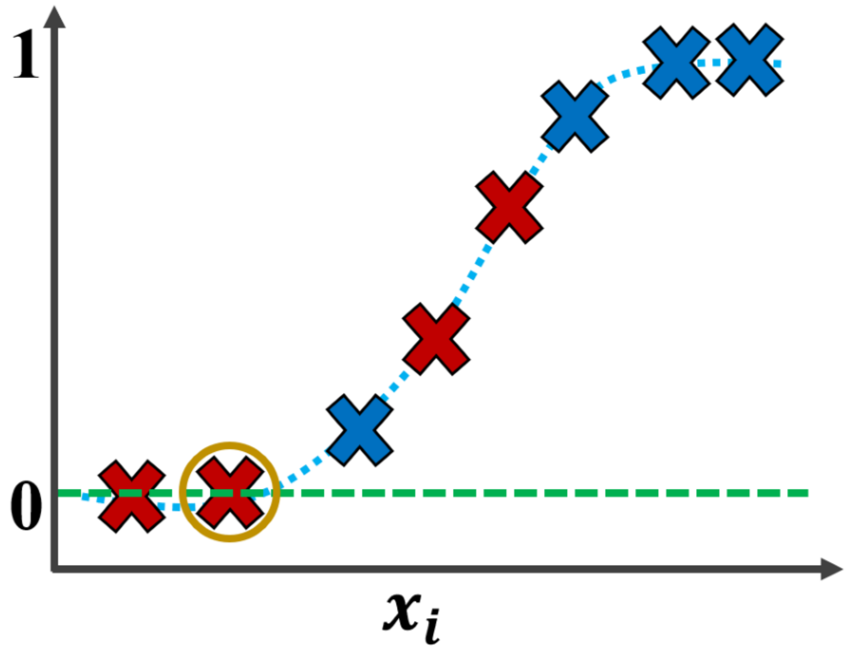
$$R_1(\rho) = \{\mathbf{x}_i \in D : S(\mathbf{x}_i) > \rho\}$$

The corresponding TPR and FPR can then be calculated.

Receiver Operating Characteristics (ROC) Analysis

$$\rho^{\min} = \min_i\{S(\mathbf{x}_i)\}, \quad \rho^{\max} = \max_i\{S(\mathbf{x}_i)\}$$

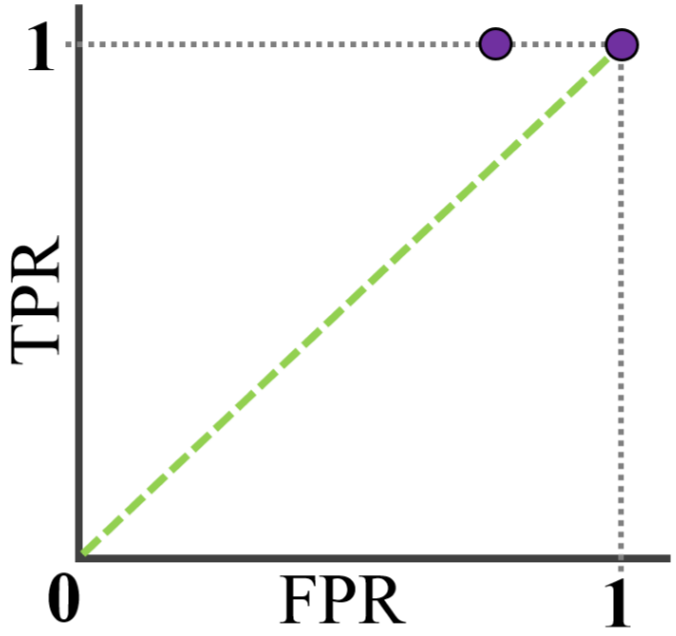
For ρ in $[\rho^{\min}, \rho^{\max}]$, the set of positive points are $R_1(\rho) = \{\mathbf{x}_i \in D : S(\mathbf{x}_i) > \rho\}$. The corresponding TPR and FPR can then be calculated.



	Pred P	Pred N
GT P	4	0
GT N	3	1

$$TPR = \frac{4}{4+0} = 1$$

$$FPR = \frac{3}{3+1} = 0.75$$



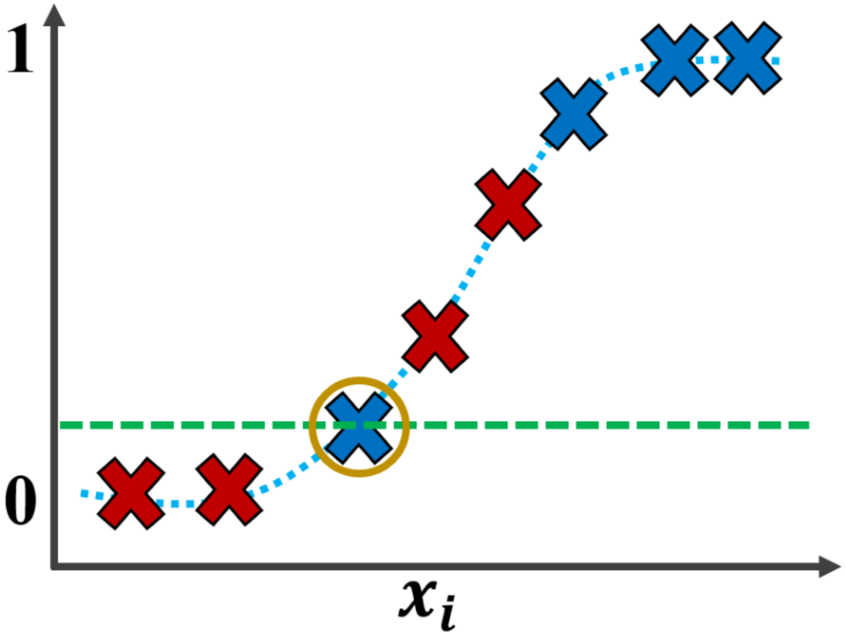
Example from: Statquest with Josh Stramer, https://www.youtube.com/watch?v=4jRBRDbJemM

Receiver Operating Characteristics (ROC) Analysis

$$\rho^{\min} = \min_i \{S(\mathbf{x}_i)\}, \quad \rho^{\max} = \max_i \{S(\mathbf{x}_i)\}$$

For ρ in $[\rho^{\min}, \rho^{\max}]$, the set of positive points are $R_1(\rho) = \{\mathbf{x}_i \in D : S(\mathbf{x}_i) > \rho\}$.

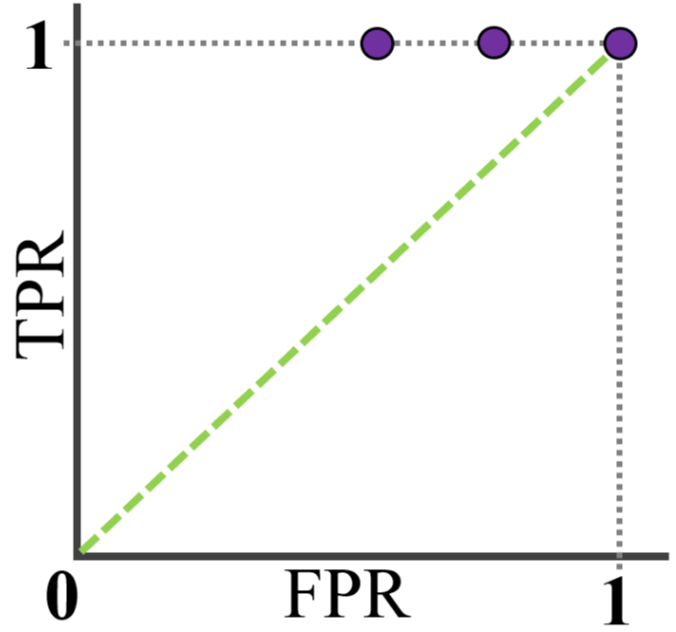
The corresponding TPR and FPR can then be calculated.



	Pred P	Pred N
GT P	4	0
GT N	2	2

$$TPR = \frac{4}{4+0} = 1$$

$$FPR = \frac{2}{2+2} = 0.5$$

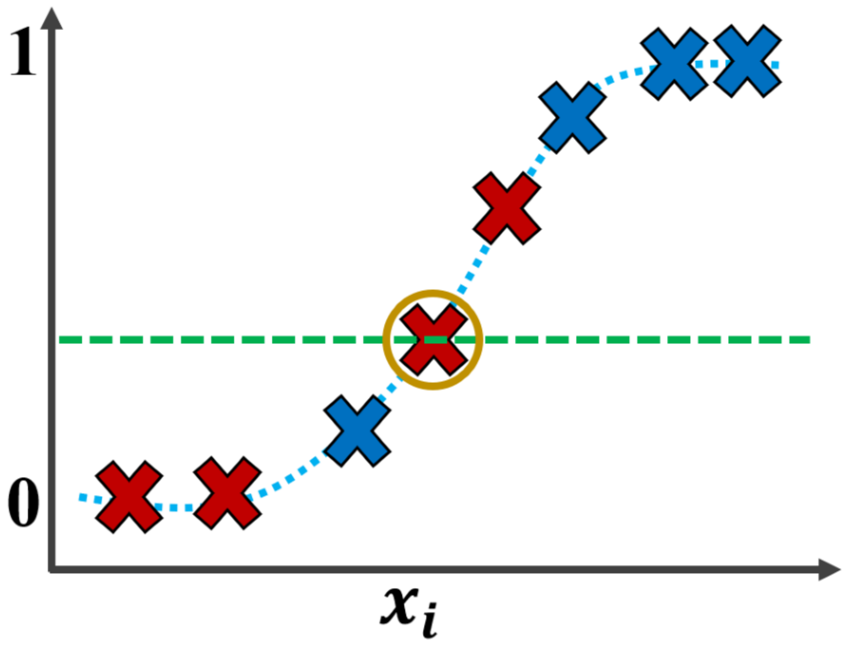


Example from: Statquest with Josh Stramer, https://www.youtube.com/watch?v=4jRBRDbJemM

Receiver Operating Characteristics (ROC) Analysis

$$\rho^{\min} = \min_i \{S(\mathbf{x}_i)\}, \quad \rho^{\max} = \max_i \{S(\mathbf{x}_i)\}$$

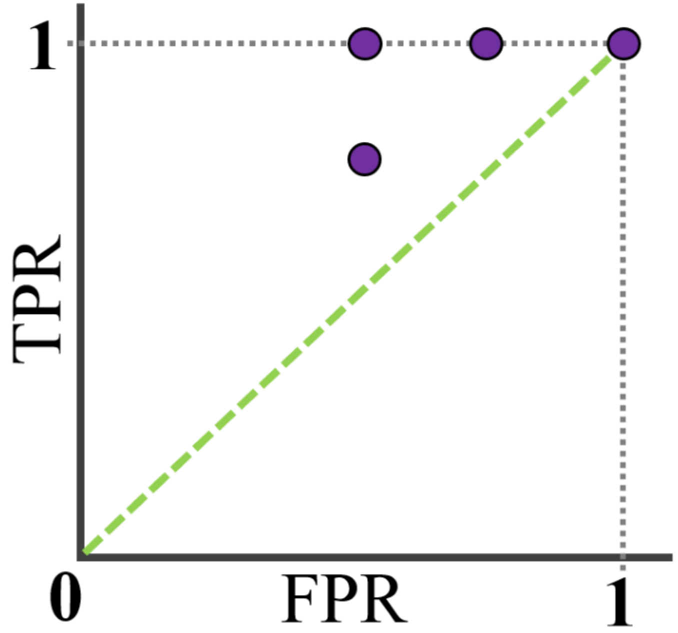
For ρ in $[\rho^{\min}, \rho^{\max}]$, the set of positive points are $R_1(\rho) = \{\mathbf{x}_i \in D : S(\mathbf{x}_i) > \rho\}$. The corresponding TPR and FPR can then be calculated.



	Pred P	Pred N
GT P	3	1
GT N	2	2

$$TPR = \frac{3}{3+1} = 0.75$$

$$FPR = \frac{2}{2+2} = 0.5$$

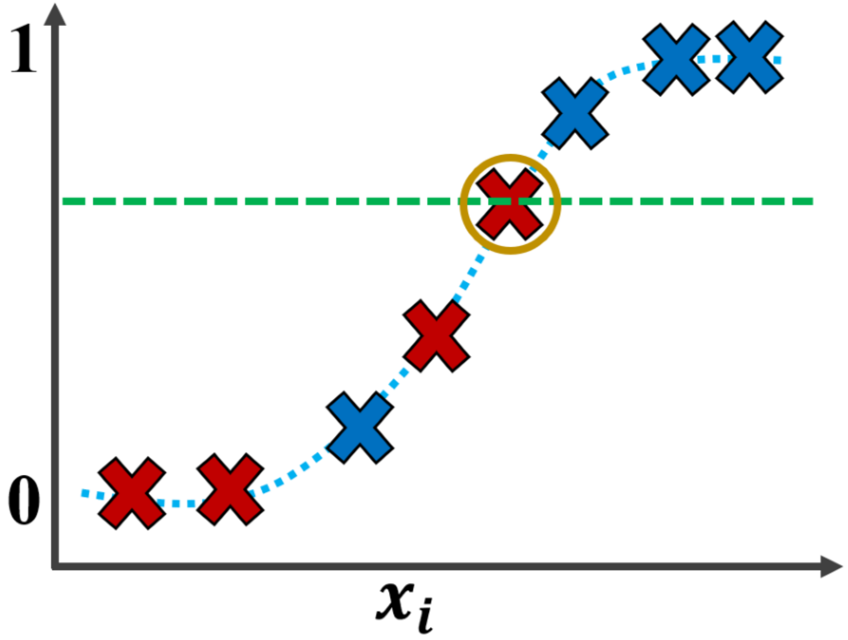


Example from: Statquest with Josh Stramer, https://www.youtube.com/watch?v=4jRBRDbJemM

Receiver Operating Characteristics (ROC) Analysis

$$\rho^{\min} = \min_i \{S(\mathbf{x}_i)\}, \quad \rho^{\max} = \max_i \{S(\mathbf{x}_i)\}$$

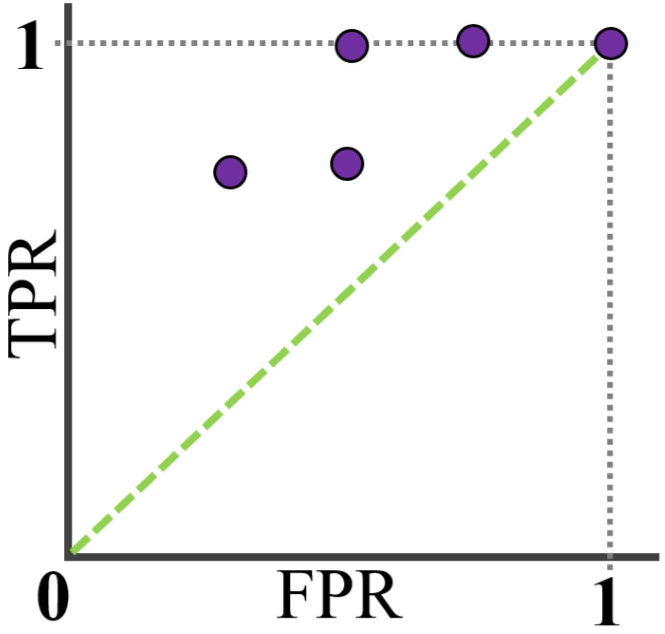
For ρ in $[\rho^{\min}, \rho^{\max}]$, the set of positive points are $R_1(\rho) = \{\mathbf{x}_i \in D : S(\mathbf{x}_i) > \rho\}$.
The corresponding TPR and FPR can then be calculated.



	Pred P	Pred N
GT P	3	1
GT N	1	3

$$TPR = \frac{3}{3+1} = 0.75$$

$$FPR = \frac{1}{1+3} = 0.25$$

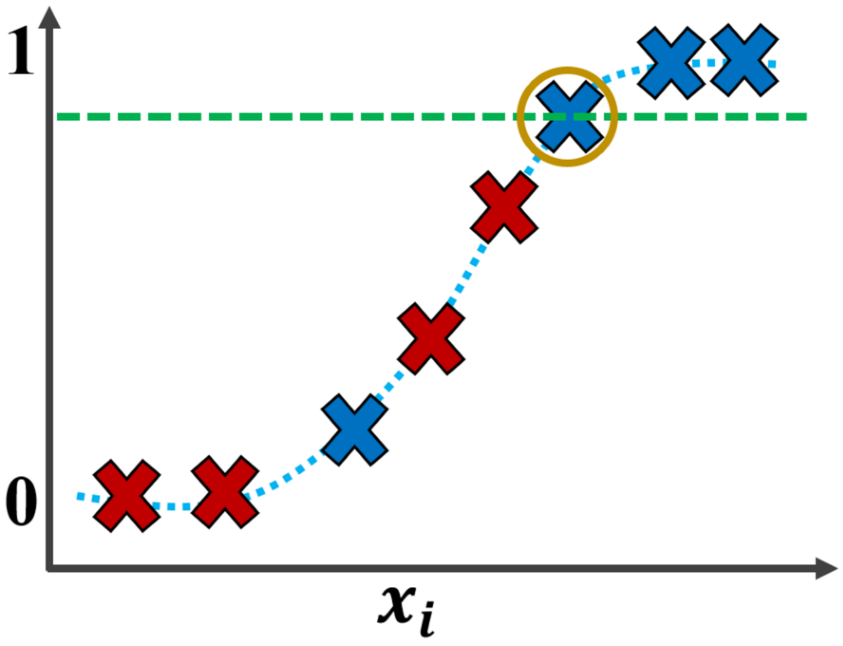


Example from: Statquest with Josh Stramer, https://www.youtube.com/watch?v=4jRBRDbJemM

Receiver Operating Characteristics (ROC) Analysis

$$\rho^{\min} = \min_i \{S(\mathbf{x}_i)\}, \quad \rho^{\max} = \max_i \{S(\mathbf{x}_i)\}$$

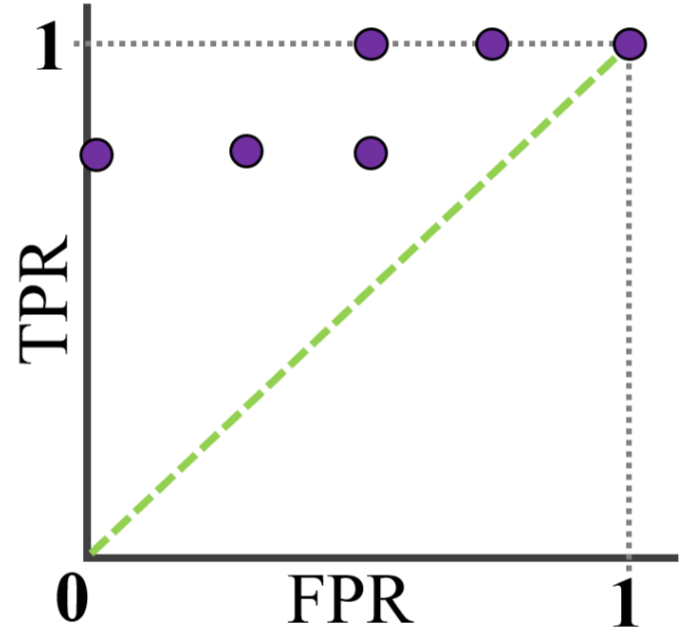
For ρ in $[\rho^{\min}, \rho^{\max}]$, the set of positive points are $R_1(\rho) = \{\mathbf{x}_i \in D : S(\mathbf{x}_i) > \rho\}$.
The corresponding TPR and FPR can then be calculated.



	Pred P	Pred N
GT P	3	1
GT N	0	4

$$TPR = \frac{3}{3+1} = 0.75$$

$$FPR = \frac{0}{0+4} = 0$$

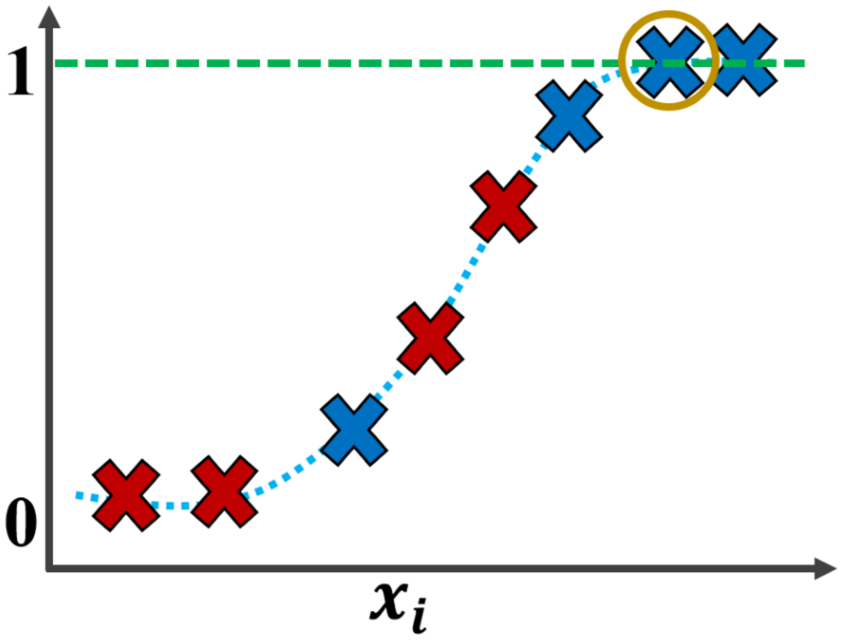


Example from: Statquest with Josh Stramer, https://www.youtube.com/watch?v=4jRBRDbJemM

Receiver Operating Characteristics (ROC) Analysis

$$\rho^{\min} = \min_i\{S(\mathbf{x}_i)\}, \quad \rho^{\max} = \max_i\{S(\mathbf{x}_i)\}$$

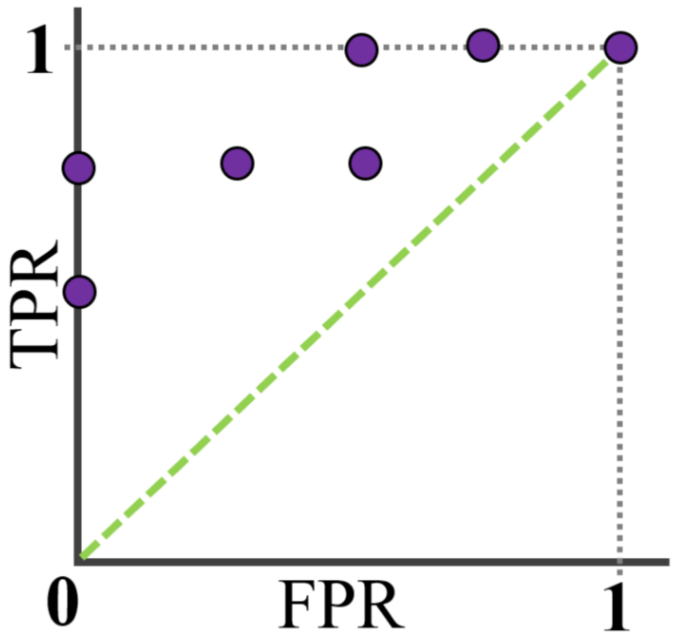
For ρ in $[\rho^{\min}, \rho^{\max}]$, the set of positive points are $R_1(\rho) = \{\mathbf{x}_i \in D : S(\mathbf{x}_i) > \rho\}$.
The corresponding TPR and FPR can then be calculated.



	Pred P	Pred N
GT P	2	2
GT N	0	4

$$TPR = \frac{2}{2+2} = 0.5$$

$$FPR = \frac{0}{0+4} = 0$$

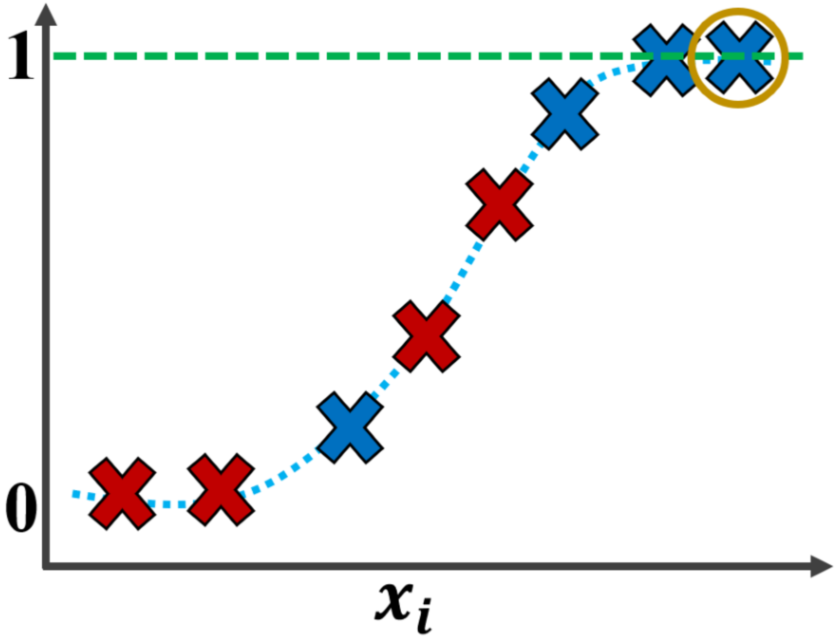


Example from: Statquest with Josh Stramer, https://www.youtube.com/watch?v=4jRBRDbJemM

Receiver Operating Characteristics (ROC) Analysis

$$\rho^{\min} = \min_i \{S(\mathbf{x}_i)\}, \quad \rho^{\max} = \max_i \{S(\mathbf{x}_i)\}$$

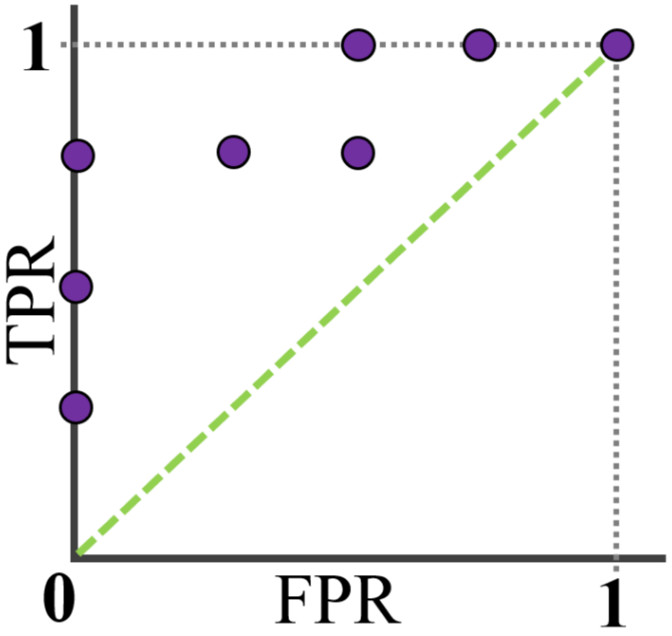
For ρ in $[\rho^{\min}, \rho^{\max}]$, the set of positive points are $R_1(\rho) = \{\mathbf{x}_i \in D : S(\mathbf{x}_i) > \rho\}$.
The corresponding TPR and FPR can then be calculated.



	Pred P	Pred N
GT P	1	3
GT N	0	4

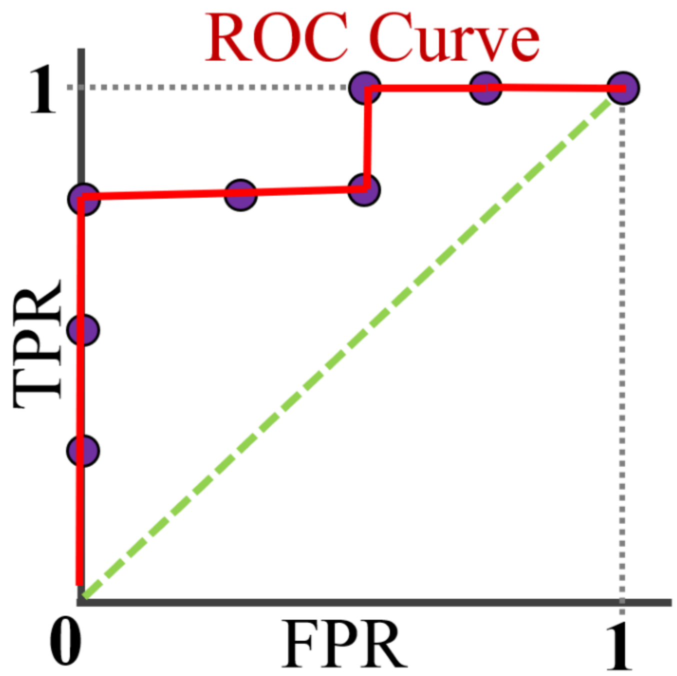
$$TPR = \frac{1}{1+3} = 0.25$$

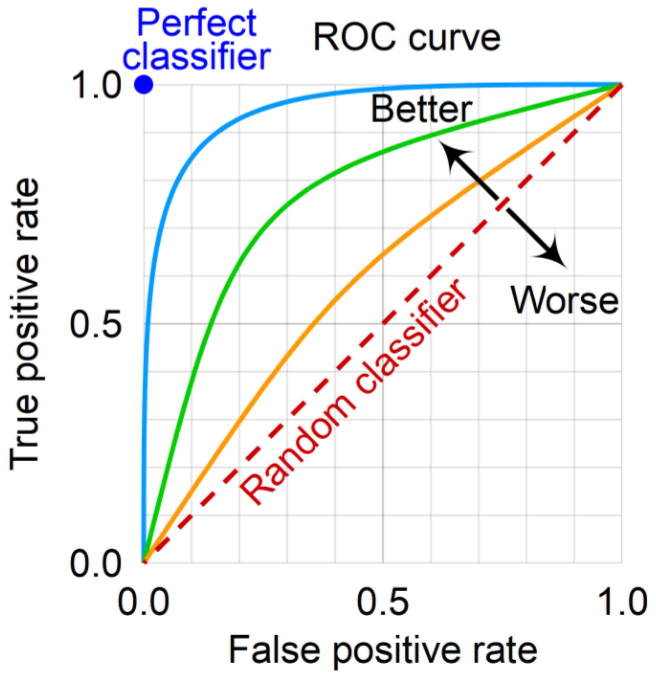
$$FPR = \frac{0}{0+4} = 0$$



Example from: Statquest with Josh Stramer, https://www.youtube.com/watch?v=4jRBRDbJemM

Receiver Operating Characteristics (ROC) Analysis





An ROC curve closer to the ideal case (top left corner) is better.

Area Under the ROC Curve (AUC): The total area of the ROC plot is 1, and therefore the AUC lies in the interval $[0, 1]$.

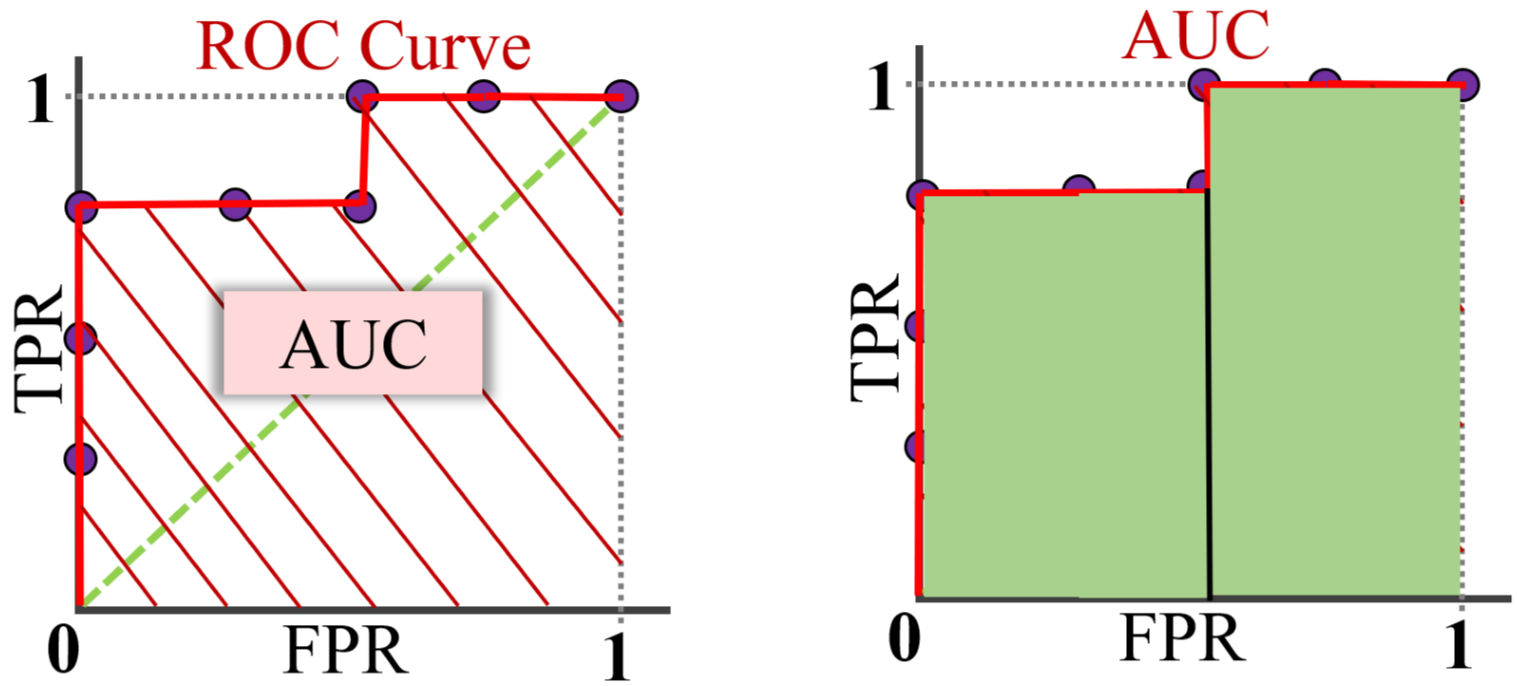
AUC is interpreted as the probability that a random positive instance will be ranked higher than a random negative instance.

Image Source: `https://upload.wikimedia.org/wikipedia/commons/1/13/Roc_curve.svg`

Area Under the ROC Curve (AUC)

The total area of the ROC plot is 1, and therefore the AUC lies in the interval $[0, 1]$.

AUC is interpreted as the probability that a random positive instance will be ranked higher than a random negative instance.



The AUC can easily be calculated by breaking down the overall region into (1) rectangles, and/or (ii) trapezoids.

k-Fold Cross Validation

Used to eliminate the chance of a model being trained and evaluated on one very favourable training-test split.

1. A dataset D is divided into n_f approx. equal sized *folds* $D_1, ..., D_{n_f}$.

2. Over n_f no. of *turns*, a model is fit to a training set, and then evalauted on a test set.

3. In the i-th turn, the fold D_i is treated as the test set, and the rest of the folds $D \backslash D_i$ are combined to form the training set. A performance measure E_i is evaluated on the test set D_i.

k-Fold Cross Validation

3. In the i-th turn, the fold D_i is treated as the test set, and the rest of the folds $D \backslash D_i$ are combined to form the training set. A performance measure E_i is evaluated on the test set D_i.

4. The k-fold cross validated performance is measured in terms of the mean and standard-deviation of the measured performance across all folds:

$$\mu_E = \frac{1}{|n_f|} \sum_{i=1}^{n_f} E_i,$$

$$\sigma_E = \frac{1}{|n_f|} \sum_{i=1}^{n_f} (E_i - \mu_E)^2.$$

Usually k is 5 or 10. The case of $k = n$ is called leave-one-out cross-validation.

Model-Agnostic Learning

Notations

Set of instances: X

Set of possible target concepts: C

Any target function $y = c(\mathbf{x}), c \in C$

Set of hypotheses: H

Any learnable function $\hat{y} = h(\mathbf{x}), h \in H$

A learner observes a sequence D of training examples $< \mathbf{x}, c(\mathbf{x}) >$, $c \in C$.

No Free Lunch Theorem

Notations:

Let $P(h)$ be the probability that an algorithm will produce hypothesis h after training.

Let $P(h|D)$ be the probability that an algorithm will produce hypothesis h after training on dataset D.

For a general loss function L, let $E = L$ be the scalar error or cost.

The expected error given dataset D:

$$\mathbb{E}[E|D] = \sum_{c} \sum_{h} \sum_{x \neq D} [1 - \delta(c(x), h(x))] P(x) P(h|D) P(c|D)$$

Without prior knowledge of $P(c|D)$, it is difficult to prove the generalization performance of any learning algorithm $P(h|D)$.

The expected generalization error given a true concept $c(x)$ and some candidate learning algorithms is $P_k(h(x)|D)$:

$$\mathbb{E}_k[E|c, D] = \sum_{x \neq D} [1 - \delta(c(x), h(x))] P(x) P_k(h|D)$$

No Free Lunch Theorem

For any two learning algorithms $P_1(h|D)$ and $P_2(h|D)$, the following are true, independent of the sampling distribution $P(x)$ and the number of training points $|D| = n$:

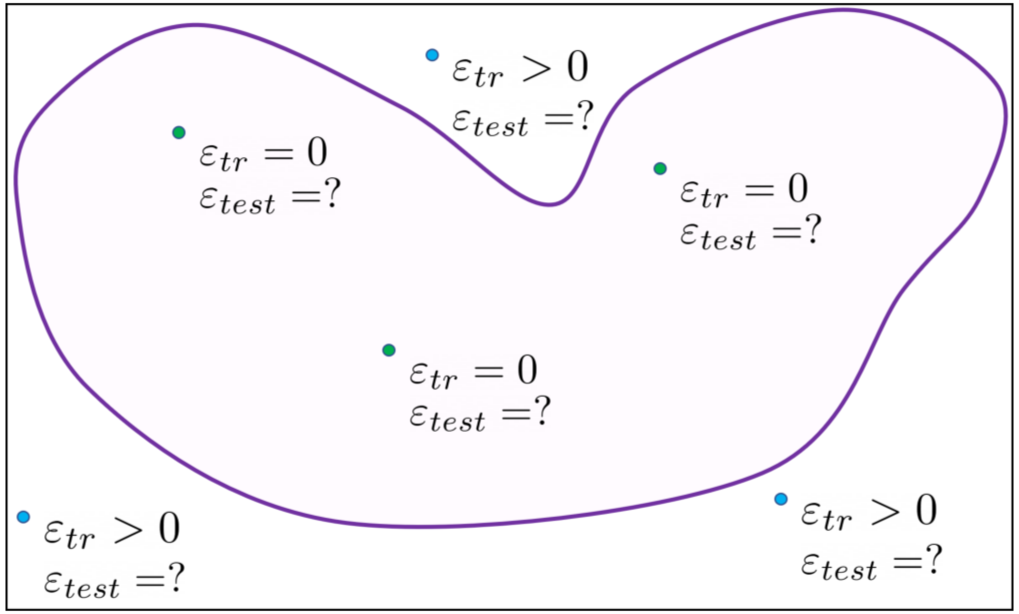
1. Uniformly averaged over all target functions c, $\mathbb{E}_1[E|c, n] - \mathbb{E}_2[E|c, n] = 0$.
2. For any fixed training set D, uniformly averaged over c, $\mathbb{E}_1[E|c, D] - \mathbb{E}_2[E|c, D] = 0$.
3. Uniformly averaged over all priors $P(c)$, $\mathbb{E}_1[E|n] - \mathbb{E}_2[E|n] = 0$.
4. For any fixed training set D, uniformly averaged over all priors $P(c)$, $\mathbb{E}_1[E|D] - \mathbb{E}_2[E|D] = 0$.

PAC Learning

Can the generalization error be bound by the number of training samples?

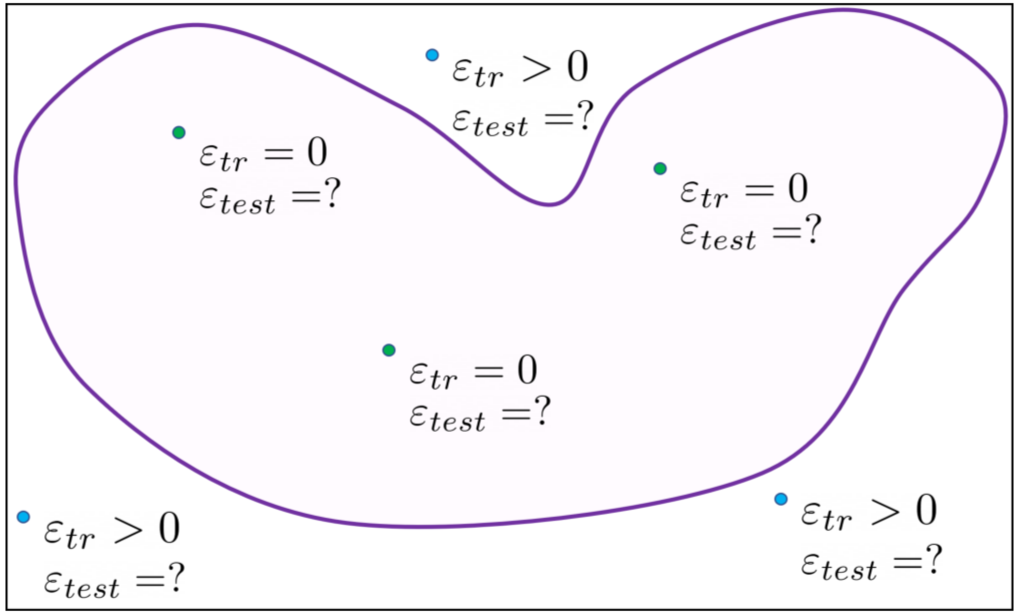
PAC Learning

Can the generalization error be bound by the number of training samples?
Version Space: Set of hypothesis that have zero training error.



PAC Learning

Can the generalization error be bound by the number of training samples?
Version Space: Set of hypothesis that have zero training error.



Theorem: (Valiant, 1984) If the hypothesis space H is finite, and D is a sequence of $n \geq 1$ independent random examples of some target concept c, then for any $0 \leq \varepsilon \leq 1$, the probability that $VS_{H,D}$ contains a hypothesis with error greater than ε is less than $|H|e^{-\varepsilon n}$, i.e.,

$$Pr[Err > \varepsilon] < |H|e^{-\varepsilon n}$$

PAC Learning

Theorem (Valiant, 1984): If the hypothesis space H is finite, and D is a sequence of $n \geq 1$ independent random examples of some target concept c, then for any $0 \leq \varepsilon \leq 1$, the probability that $VS_{H,D}$ contains a hypothesis with error greater than ε is less than $|H|e^{-\varepsilon n}$, i.e.,

$$Pr[Err > \varepsilon] < |H|e^{-\varepsilon n}$$

Proof:

Probability that one sample will be correctly classified $= 1 - \varepsilon$

Probability that n samples will be correctly classified $= (1 - \varepsilon)^n$

$$(1 - \varepsilon)^n \leq e^{-\varepsilon n}$$
$$(1 - \varepsilon)^n \leq e^{-\varepsilon n} \leq |H|e^{-\varepsilon n}$$

PAC Learning

Theorem (Valiant, 1984): If the hypothesis space H is finite, and D is a sequence of $n \geq 1$ independent random examples of some target concept c, then for any $0 \leq \varepsilon \leq 1$, the probability that $VS_{H,D}$ contains a hypothesis with error greater than ε is less than $|H|e^{-\varepsilon n}$, i.e.,

$$Pr[Err > \varepsilon] < |H|e^{-\varepsilon n}$$

Let us want this probability to be at most δ, i.e.,

$$|H|e^{-\varepsilon n} \leq \delta$$

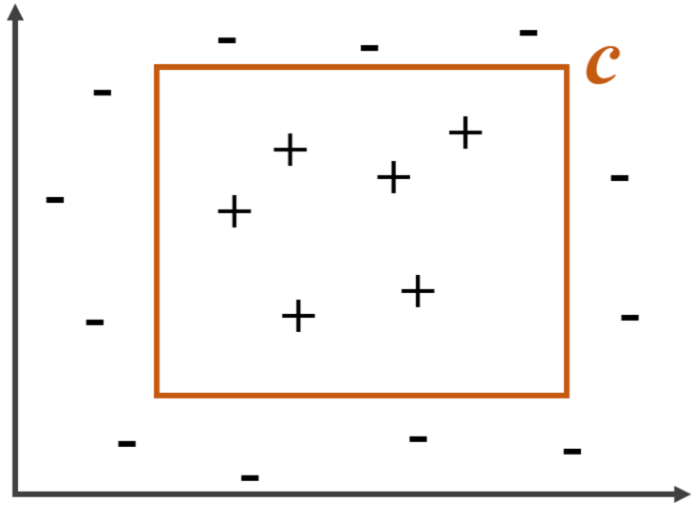
Then,

$$n \geq \frac{1}{\varepsilon}(\ln |H| + \ln(1/\delta))$$

1. With linear increase in data, the bound becomes exponentially better.
2. $|H|$ can be large, requiring more data (If $|H|$ is infinity, the bound does not help).

Example: PAC bounds - (1)

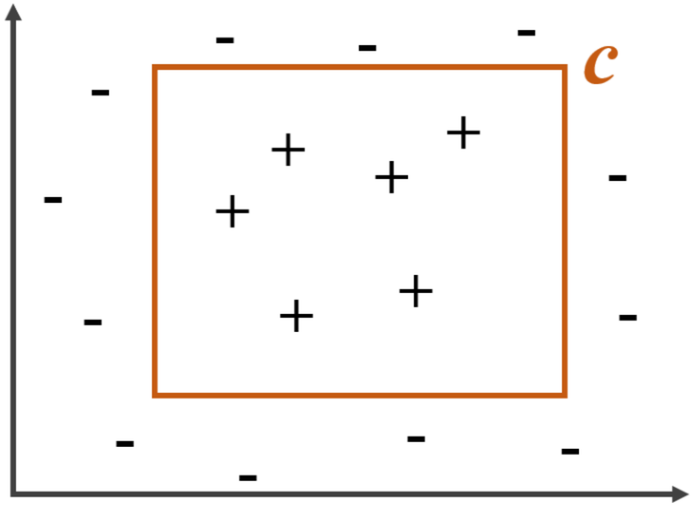
Let our instances lie in $\mathbb{R}^2$, and the target concept is known to be a rectangle with length and width parallel to the two axes.



We wish to find a bound on the number of instaces required to learn a hypothesis with error ε.

Example: PAC bounds - (1)

Let our instances lie in $\mathbb{R}^2$, and the target concept is known to be a rectangle with length and width parallel to the two axes.



We wish to find a bound on the number of instaces required to learn a hypothesis with error ε.

Let our training algorithm to learn a hypothesis be the following:

1. If there are no positive instances, the learned hypothesis is null.
2. Otherwise, the learned hypothesis is the smallest rectangle that contain all positive instances.

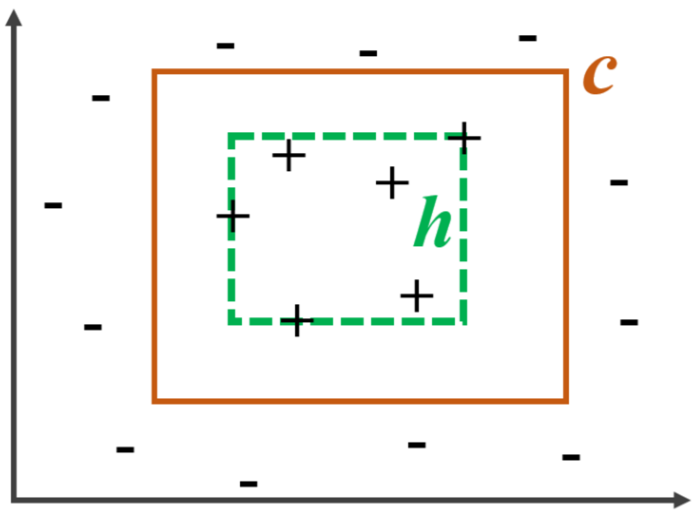
Example: PAC bounds - (2)

Let our instances lie in $\mathbb{R}^2$, and the target concept is known to be a rectangle with length and width parallel to the two axes.
We wish to find a bound on the number of instaces required to learn a hypothesis with error ε.

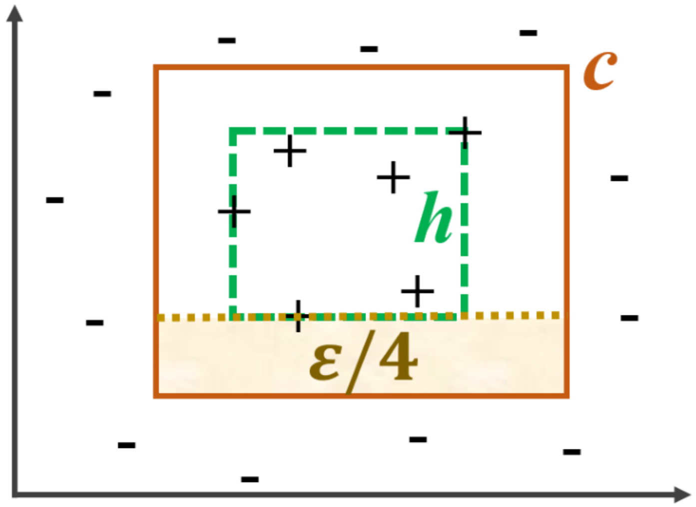
Let our training algorithm to learn a hypothesis be the following:

1. If there are no positive instances, the learned hypothesis is null.

2. Otherwise, the learned hypothesis is the smallest rectangle that contain all positive instances.



Example: PAC bounds - (3)



Let the area of the difference of rectangles be ε. A pessimistic estimate of each overlapped rectange strip $= \varepsilon/4$.

Probability that one instance will be outside the strip $= 1 - \varepsilon/4$.

Probability that n instances will be outside the strip $= (1 - \varepsilon/4)^n$.

Probability that n instances will be outside at least one of the four strips $= 4(1 - \varepsilon/4)^n$.

Example: PAC bounds - (4)

Probability that n instances will be outside at least one of the four strips $= 4(1 - \varepsilon/4)^n$.

Therefore,

$$4(1 - \varepsilon/4)^n < \delta$$
$$\implies n > \ln(\delta/4)/\ln(1 - \varepsilon/4)$$

For $y < 1$: $-ln(1 - y) = y + y^2/2 + y^3/3 + \ldots$

$$\implies 1 - y < e^{-y}$$

Hence, $n > \frac{4}{\varepsilon} \ln \frac{4}{\delta}$.