Decision Theory

Performance Evaluation → Application

Accuracy
(performance metrics)

Computational efficiency

Interpretability

Deciding how to operate our algorithms in practice

(after we've evaluated generalization performance)

Time to make a decision…

Exercise inspired by Mausam, University of Washington, CSE573

State of Nature

Action	Poor market performance **Payoff**	Good market performance **Payoff**	
Buy Apple	-1,000	1,700	-10% to +17% return
Buy Google	-2,000	2,100	-20% to +21% return
Buy bonds	500	500	Guaranteed 5% return

How to invest $10,000?

Maximax

Optimism

Select the maximum of the maximum payoff

Action	State of Nature: Poor market performance Payoff	State of Nature: Good market performance Payoff	Criterion: Maximum payoff for an action
Buy Apple	-1,000	1,700	1,700
Buy Google	-2,000	2,100	2,100 ← **Maximax**
Buy bonds	500	500	500

Maximin

Pessimism

Select the maximum of the minimum payoffs

Action	State of Nature: Poor market performance Payoff	State of Nature: Good market performance Payoff	Criterion: Minimum payoff for an action
Buy Apple	-1,000	1,700	-1,000
Buy Google	-2,000	2,100	-2,000
Buy bonds	500	500	500

← **Maximin**

Minimax

Select the minimum maximum regret

		State of Nature				Criterion
		Poor market performance		Good market performance		Maximum regret for an action
		Payoff	Regret	Payoff	Regret	
Action	Buy Apple	-1,000	1,500	1,700	400	1,500 ← **Minimax**
	Buy Google	-2,000	2,500	2,100	0	2,500
	Buy bonds	500	0	500	1,600	1,600

Which decision would I regret least?

Regret = Opportunity Loss
Difference between a decision made and an optimal decision

Next: factor in probabilities of different outcomes

Expected Payoff: Equal likelihood

	State of Nature Poor market performance **Payoff**	**State of Nature** Good market performance **Payoff**	**Criterion** Expected reward/ payoff
Action Buy Apple	-1,000	1,700	350
Buy Google	-2,000	2,100	50
Buy bonds	500	500	500
State Probability:	0.5	0.5	

Select the highest average payoff ASSUMING all states are of equal probability

Maximum Expected Reward

Expected Payoff

		State of Nature		Criterion
		Poor market performance **Payoff**	Good market performance **Payoff**	Expected reward/ payoff
Action	Buy Apple	-1,000	1,700	890
	Buy Google	-2,000	2,100	870
	Buy bonds	500	500	500
State Probability:		0.3	0.7	

Select the highest average payoff assuming state probabilities from prior knowledge

Maximum Expected Reward

Decision making design pattern

1. Define a measure of risk or reward

2. Select the action that optimizes that metric

Notation

$EV(a_i) = V(a_i|s_0)P(s_0) + V(a_i|s_1)P(s_1)$

Expected reward / payoff

State of Nature (s)

	Poor market performance $s = s_0$	Excellent market performance $s = s_1$	**Expected Reward** $EV(a_i)$
Buy Apple $a = a_0$	$V(a_0\|s_0)$ -1,000	$V(a_0\|s_1)$ 1,700	$(0.3)(-1000) + (0.7)(1700)$ $= \mathbf{890}$
Buy Google $a = a_1$	$V(a_1\|s_0)$ -2,000	$V(a_1\|s_1)$ 2,100	$(0.3)(-2000) + (0.7)(2100)$ $= \mathbf{870}$
Buy bonds $a = a_2$	$V(a_2\|s_0)$ 500	$V(a_2\|s_1)$ 500	$(0.3)(500) + (0.7)(500)$ $= \mathbf{500}$

Action

State Probability: $P(s_0) =$ 0.3 $P(s_1) =$ 0.7

Risk = expected loss
(cost)

Loss: $\lambda(a_i|s_j) \triangleq$ Loss incurred by choosing action i and the state of nature being state j

Risk:
Expected loss

$$R(a_i) = \sum_{j=1}^{N_s} \lambda(a_i|s_j)P(s_j)$$

Goal: Select action i for which $R(a_i)$ is minimum

Payoff

State of Nature

Action	Poor market performance	Good market performance
Buy Apple	-1,000	1,700
Buy Google	-2,000	2,100
Buy bonds	500	500

Loss

(here we define loss in terms of opportunity cost)

State of Nature

Action	Poor market performance	Good market performance
Buy Apple	1,500	400
Buy Google	2,500	0
Buy bonds	0	1,600

Investments: loss

$R(a_i) = \lambda(a_i|s_0)P(s_0) + \lambda(a_i|s_1)P(s_1)$

Risk (Expected loss)

State of Nature (s)

Action	Poor market performance $s = s_0$	Excellent market performance $s = s_1$	**Risk** (Expected Loss) $R(a_i)$
Buy Apple $a = a_0$	$\lambda(a_0\|s_0)$ 1,500	$\lambda(a_0\|s_1)$ 400	$(0.3)(1500) + (0.7)(400)$ $= \mathbf{730}$
Buy Google $a = a_1$	$\lambda(a_1\|s_0)$ 2,500	$\lambda(a_1\|s_1)$ 0	$(0.3)(2500) + (0.7)(0)$ $= \mathbf{750}$
Buy bonds $a = a_2$	$\lambda(a_2\|s_0)$ 0	$\lambda(a_2\|s_1)$ 1,600	$(0.3)(0) + (0.7)(1600)$ $= \mathbf{1220}$

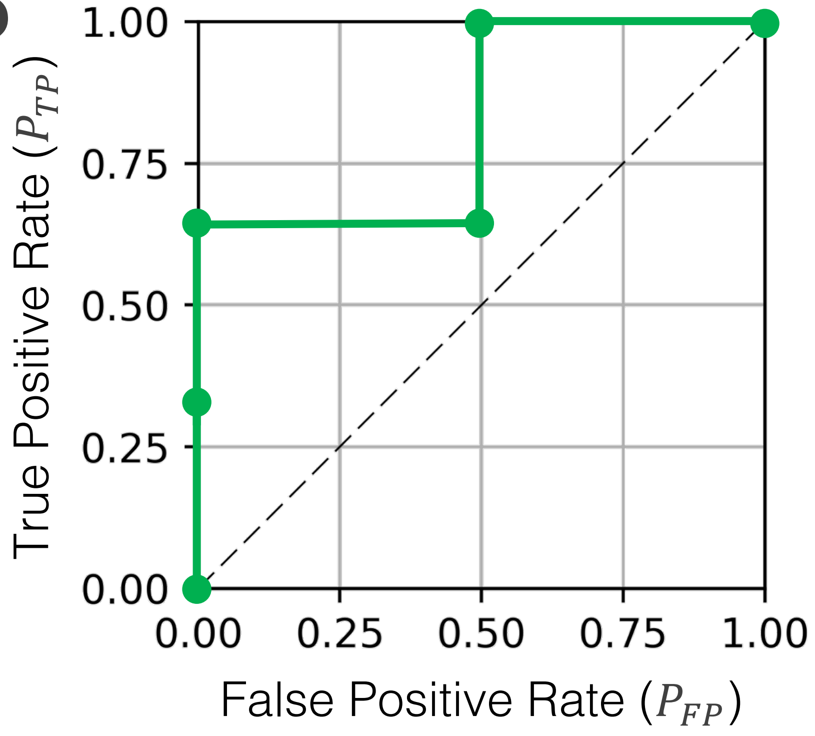
State Probability: $P(s_0) =$ 0.3 $P(s_1) =$ 0.7

We can use risk to choose where to operate along an ROC curve

Where to operate along ROC?

State of Nature

	Class 0	Class 1
Estimate Class 0	$\lambda_{00} = \mathbf{0}$	$\lambda_{01} = \mathbf{100}$ **False negative**
Estimate Class 1	$\lambda_{10} = \mathbf{1}$ **False positive**	$\lambda_{11} = \mathbf{0}$



$$\lambda_{ij} = \lambda(a_i|s_j)$$

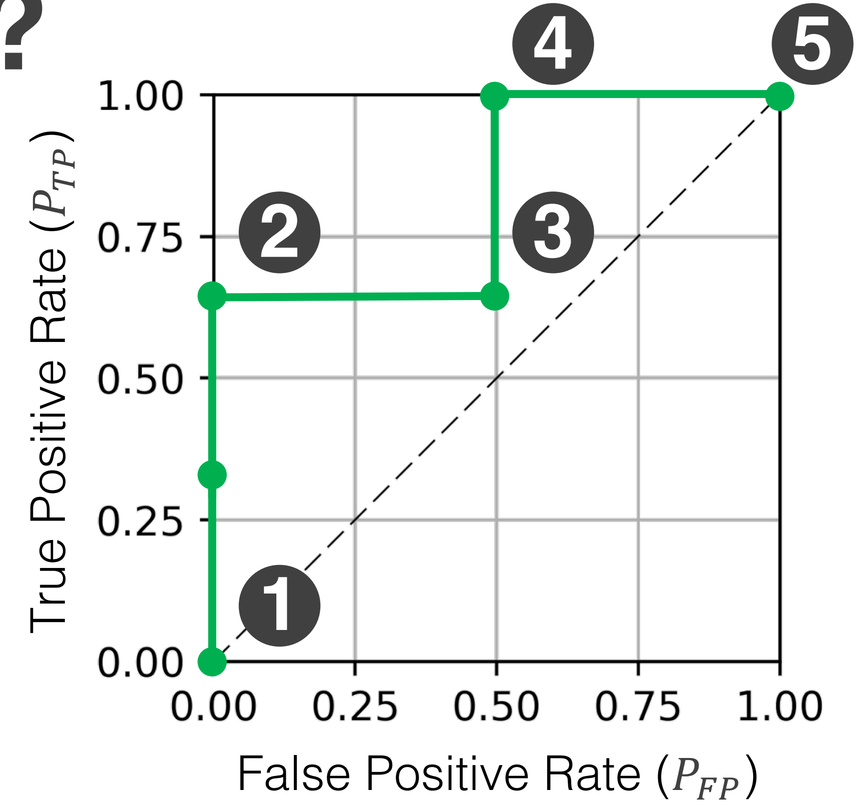
Loss from classifying as class i when state of nature is class j

NOTE: Actions, a_i, are choices of points to operate at along the ROC curve (threshold values of the confidence score)

- Assume our classification problem is landmine detection
- A false positive wastes some time and resources, but a false negative may cost a life

Where to operate along ROC?

Action: select operating point i	Probability of false positive P_{FP}	Probability of false negative $(1 - P_{TP})$	Risk $R(a_i)$
1	0	1	100



State of Nature

Estimate		Class 0	Class 1
	Class 0	$\lambda_{00} = 0$	$\lambda_{01} = 100$
	Class 1	$\lambda_{10} = 1$	$\lambda_{11} = 0$

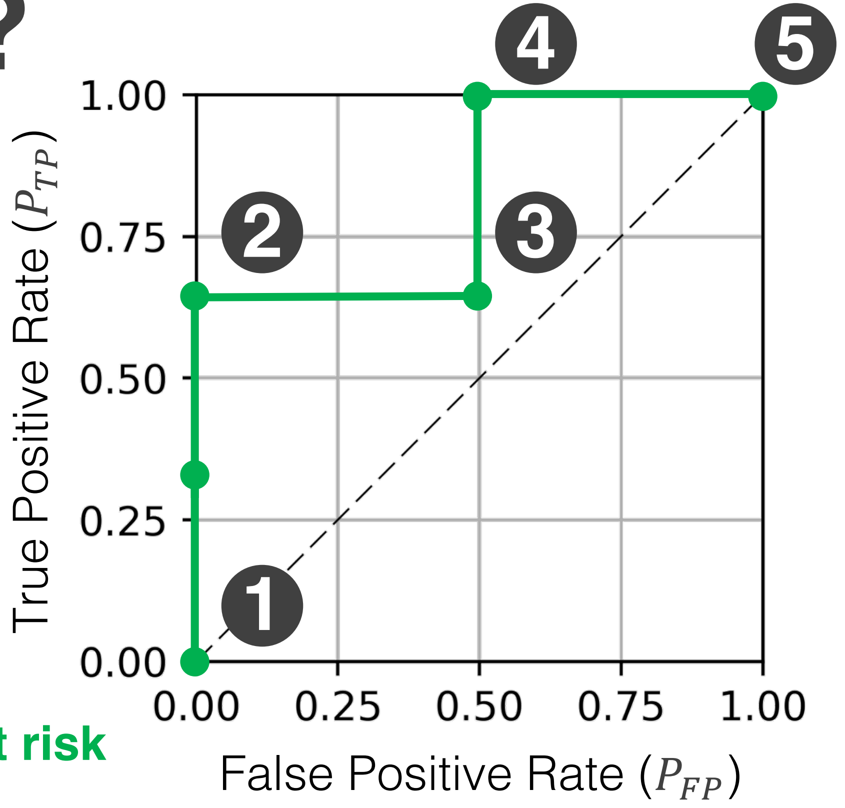
$$R(a_i) = \sum_{j=1}^{N_s} \lambda(a_i|s_j)P(s_j)$$

$$R(a_i) = \lambda_{10}\underbrace{P_{FP}(i)}_{\text{Prob of false positive}} + \lambda_{01}\underbrace{(1 - P_{TP}(i))}_{\text{Prob of false negative}}$$

Where to operate along ROC?

Action: select operating point i	Probability of false positive P_{FP}	Probability of false negative $(1 - P_{TP})$	Risk $R(a_i)$
1	0	1	100
2	0	0.33	33
3	0.5	0.33	33.5
4	0.5	0	0.5
5	1	0	1

Least risk



$$R(a_i) = \sum_{j=1}^{N_s} \lambda(a_i|s_j)P(s_j)$$

$$R(a_i) = \lambda_{10}\underbrace{P_{FP}(i)}_{\text{Prob of false positive}} + \lambda_{01}\underbrace{(1 - P_{TP}(i))}_{\text{Prob of missed detection}}$$

State of Nature

Estimate	Class 0	Class 1
Class 0	$\lambda_{00} = 0$	$\lambda_{01} = 100$
Class 1	$\lambda_{10} = 1$	$\lambda_{11} = 0$

Let’s generalize this to any binary classifier

This is how to pick what decision threshold to use for a binary classifier

Defining risk for binary decisions

		State of Nature	
		Class 0 $s = s_0$	Class 1 $s = s_1$
Estimate	Class 0 $a = a_0$	$\lambda(a_0\|s_0)$ λ_{00}	$\lambda(a_0\|s_1)$ λ_{01}
	Class 1 $a = a_1$	$\lambda(a_1\|s_0)$ λ_{10}	$\lambda(a_1\|s_1)$ λ_{11}

λ_{ij} = Loss when you classify as class i when state of nature is class j

NOTE: Actions, a_i, are **predictions** (estimate of what class a sample belongs to)

Probability from classifier (i.e. confidence score)

$$R(a_0|\boldsymbol{x}) = \lambda_{00}P(s_0|\boldsymbol{x}) + \lambda_{01}P(s_1|\boldsymbol{x})$$

$$R(a_1|\boldsymbol{x}) = \lambda_{10}P(s_0|\boldsymbol{x}) + \lambda_{11}P(s_1|\boldsymbol{x})$$

$$P(s_i|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|s_i)P(s_i)}{P(\boldsymbol{x})}$$

❶
Define the risk associated with each of the two actions

$$R(a_0|\boldsymbol{x}) = \lambda_{00}P(s_0|\boldsymbol{x}) + \lambda_{01}P(s_1|\boldsymbol{x})$$

$$R(a_1|\boldsymbol{x}) = \lambda_{10}P(s_0|\boldsymbol{x}) + \lambda_{11}P(s_1|\boldsymbol{x})$$

❷
Create a decision rule based on the data

If $R(a_0|\boldsymbol{x}) > R(a_1|\boldsymbol{x})$ then a_1 (decide class 1)

Else then a_0 (decide class 0)

We choose the rule to **minimize the risk**

❸
Express this rule in terms of the output from the classifier

$$\lambda_{00}P(s_0|\boldsymbol{x}) + \lambda_{01}P(s_1|\boldsymbol{x}) > \lambda_{10}P(s_0|\boldsymbol{x}) + \lambda_{11}P(s_1|\boldsymbol{x})$$

then a_1

$$\frac{P(s_1|\boldsymbol{x})}{P(s_0|\boldsymbol{x})} > \frac{\lambda_{10} - \lambda_{00}}{\lambda_{01} - \lambda_{11}}$$

then a_1

This can be applied any time we have an estimate of $P(s_i|\boldsymbol{x})$

Special case: Minimizing the misclassification rate

$$\frac{P(s_1|\boldsymbol{x})}{P(s_0|\boldsymbol{x})} > \frac{\lambda_{10} - \lambda_{00}}{\lambda_{01} - \lambda_{11}} \quad \text{then} \quad a_1$$ (decide class 1)

Assume that the loss is only for error, and it's the same for both types of error:

$$\lambda_{10} = \lambda_{01} \quad \text{and} \quad \lambda_{00} = \lambda_{11} = 0$$

Then the decision rule simplifies to the following:

$$\frac{P(s_1|\boldsymbol{x})}{P(s_0|\boldsymbol{x})} > 1 \quad \text{then} \quad a_1$$ (decide class 1)

else a_0 (decide class 0)

Pick whichever class is more likely given the data

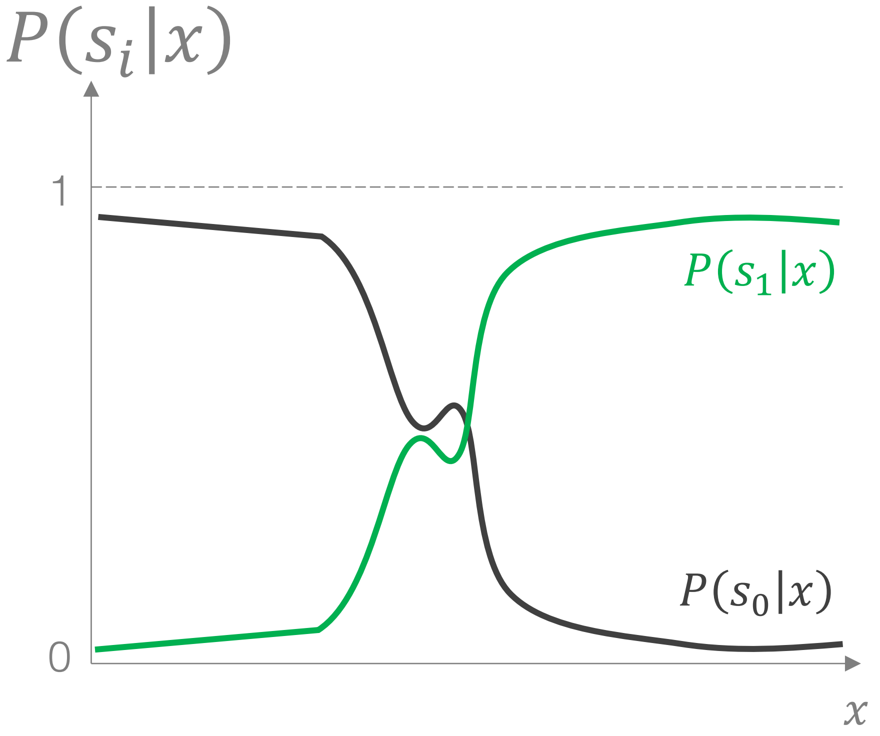
Recall Bayes' Rule

Note: The **evidence** ensures the posterior integrates to 1

$$\underbrace{P(s_i|\boldsymbol{x})}_{\text{Posterior}} = \frac{\overbrace{P(\boldsymbol{x}|s_i)}^{\text{Likelihood}}\overbrace{P(s_i)}^{\text{Prior}}}{\underbrace{P(\boldsymbol{x})}_{\text{Evidence}}}$$

Posterior

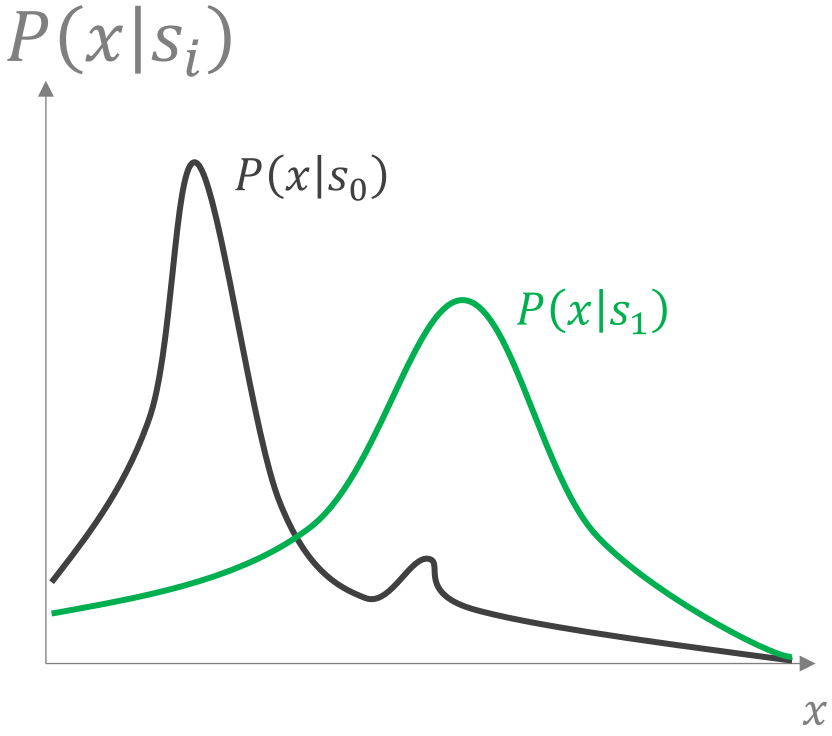
Answers the question: after seeing the data – which class is it most likely to belong to? Summing this across classes equals 1.



Discriminative models estimate this

Likelihood

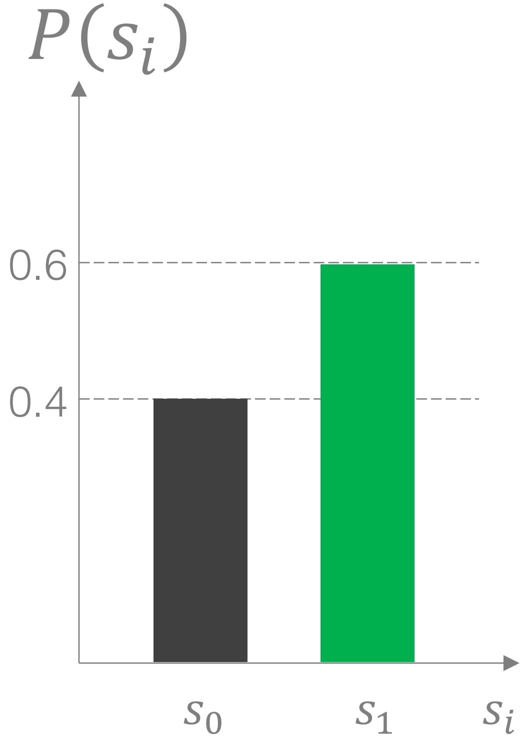
Answers the question: if I knew which class a sample belongs to, how are the data distributed?



Generative models also estimate this

Prior

Answers the question: what do I anticipate is the balance between my classes?



Likelihood ratio

Use Bayes rule to express this as a function of likelihoods

$$\frac{P(s_1|\boldsymbol{x})}{P(s_0|\boldsymbol{x})} > \frac{\lambda_{10} - \lambda_{00}}{\lambda_{01} - \lambda_{11}}$$

$$P(s_i|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|s_i)P(s_i)}{P(\boldsymbol{x})}$$

$$\frac{P(\boldsymbol{x}|s_1)P(s_1)}{P(\boldsymbol{x}|s_0)P(s_0)} > \frac{\lambda_{10} - \lambda_{00}}{\lambda_{01} - \lambda_{11}}$$

then a_1 (decide class 1)

Can easily factor in prior knowledge about the classes

The decision rule can be expressed as a **likelihood ratio**

$$\frac{P(\boldsymbol{x}|s_1)}{P(\boldsymbol{x}|s_0)} > \left(\frac{\lambda_{10} - \lambda_{00}}{\lambda_{01} - \lambda_{11}}\right)\frac{P(s_0)}{P(s_1)}$$

then a_1 (decide class 1)

else a_0 (decide class 0)

Note that this doesn't rely on posterior probabilities

This can be readily applied to **generative models**

Takeaways

To make a decision:

1. Define a measure of risk or reward
2. Select the action that optimizes that metric

Decision theory guides us in how to operate supervised learning algorithms in practice

Decision theory systematically incorporates the relative importance of different error types