

MAI 5100: Fundamentals of Artificial Intelligence

Instructor: Dr. Christopher Clarke

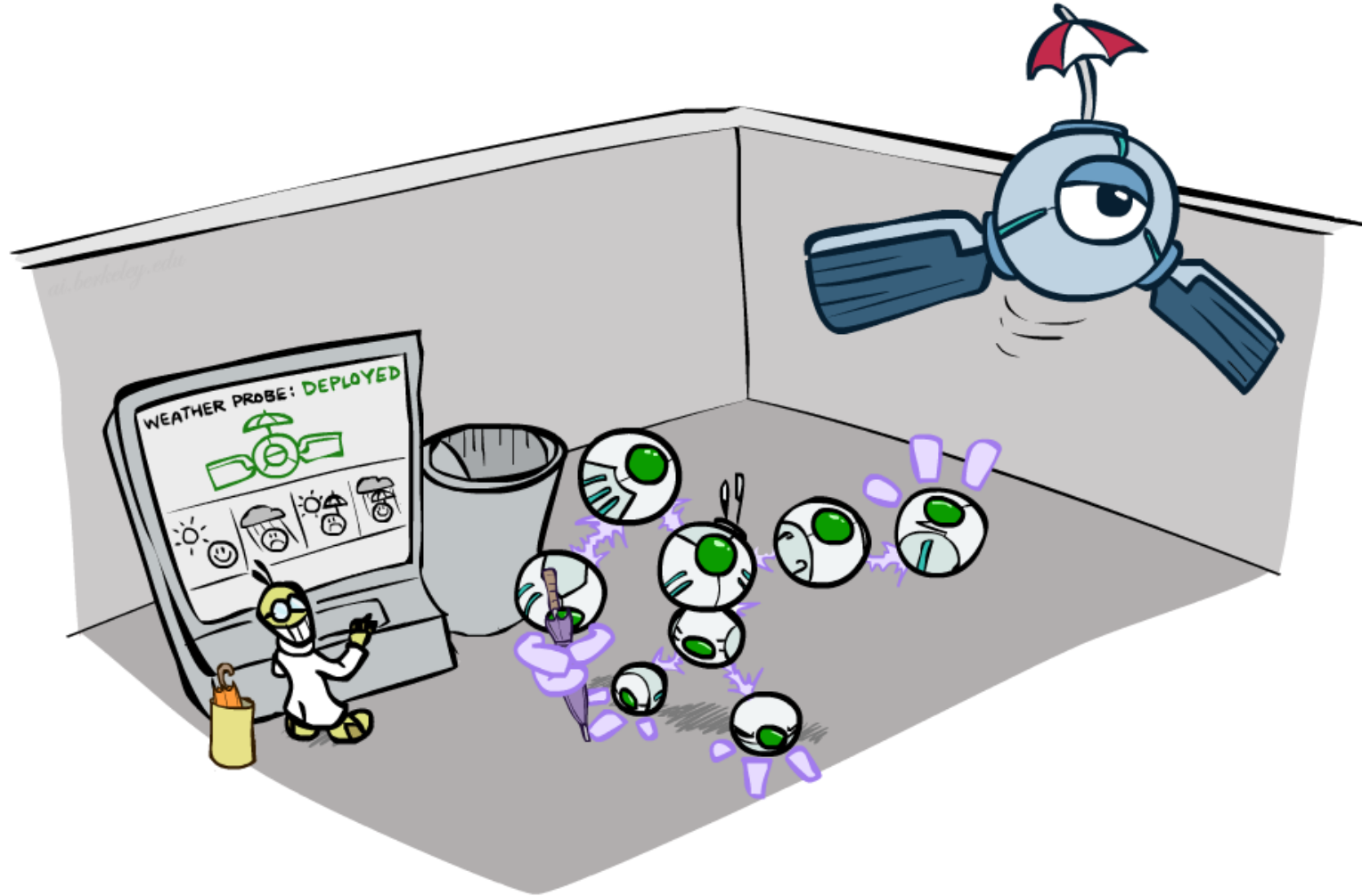
Overview & Agenda (week 14)

- **General:**
 - HW3 due on 06/18
 - Project final presentations on 06/21.
 - Final report due on 06/23.

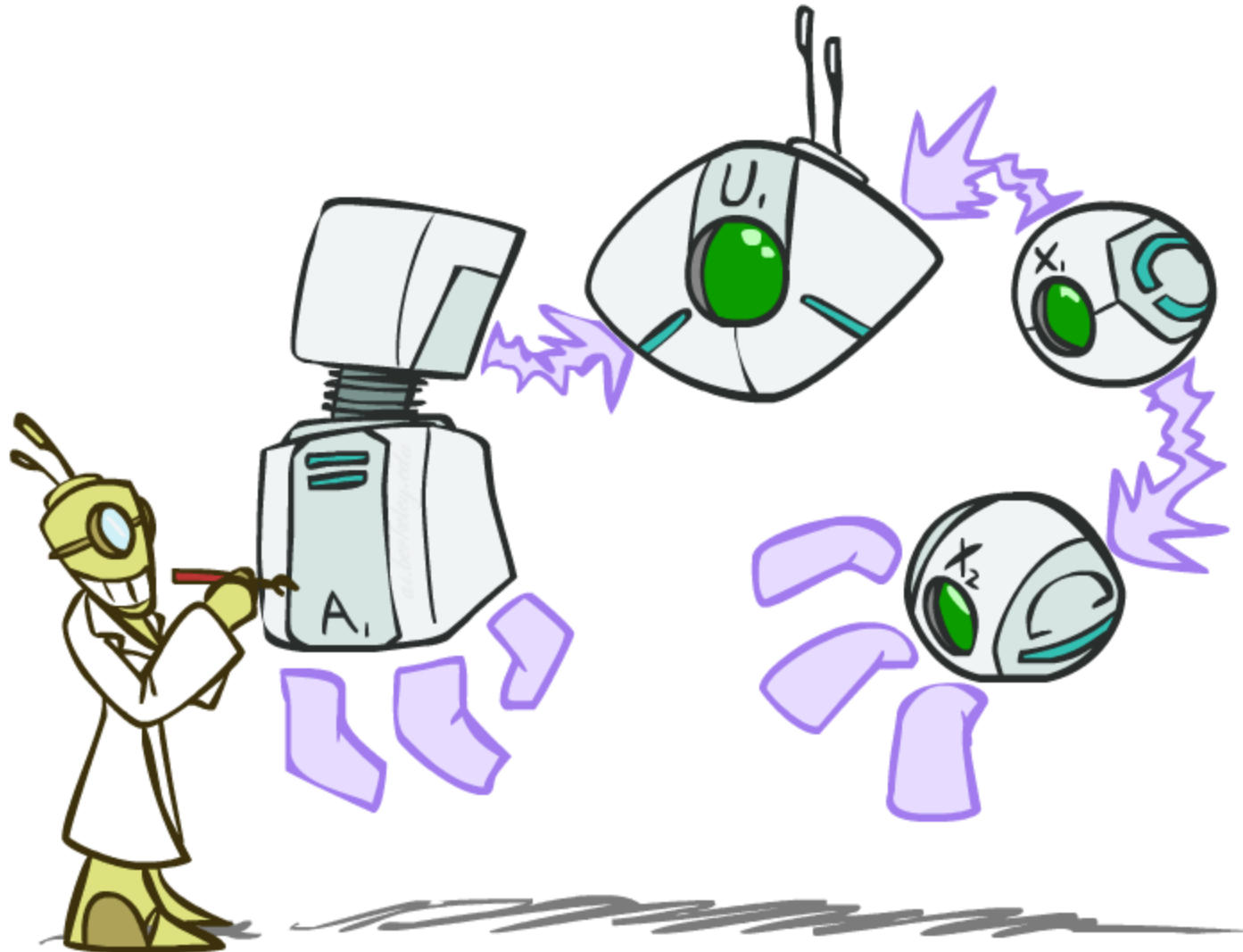
Overview & Agenda (week 14)

- **This week:**
 - Bayes Nets: Sampling
 - Decision Networks and VPI
 - Naive Bayes

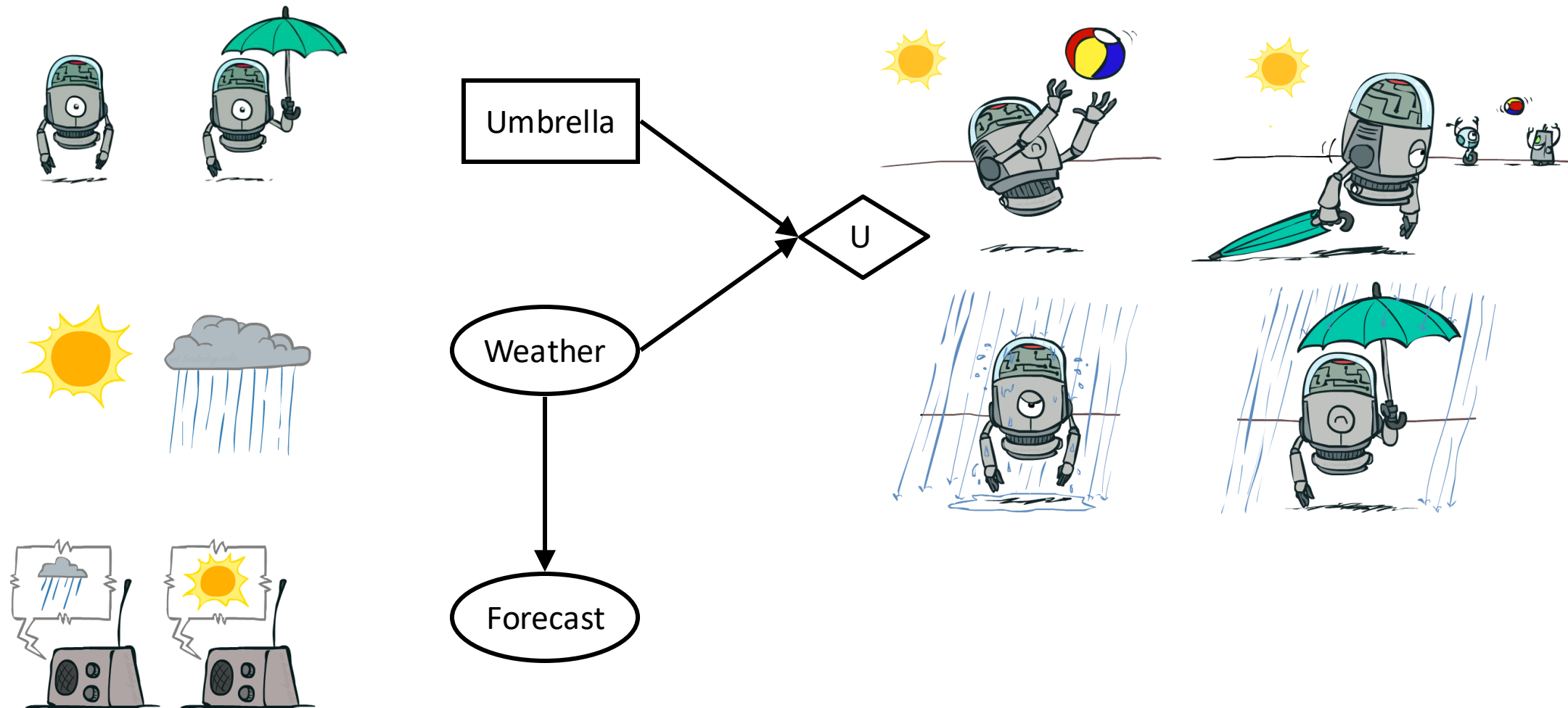
Decision Networks and Value of Perfect Information



Decision Networks



Decision Networks



Decision Networks

- **MEU: choose the action which maximizes the expected utility given the evidence**

- Can directly operationalize this with decision networks

- Bayes nets with nodes for utility and actions
- Lets us calculate the expected utility for each action

- New node types:



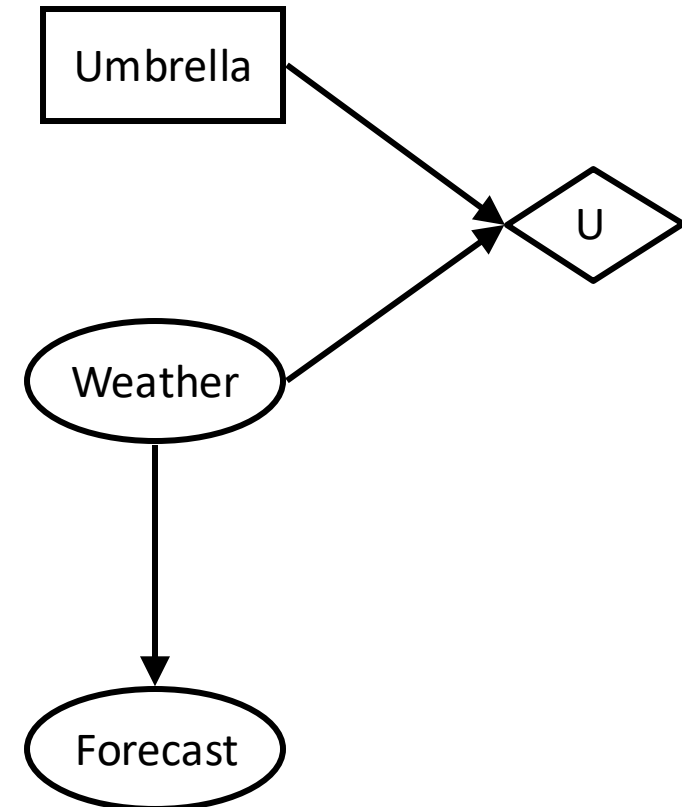
- Chance nodes (just like BNs)



- Actions (rectangles, cannot have parents, act as observed evidence)

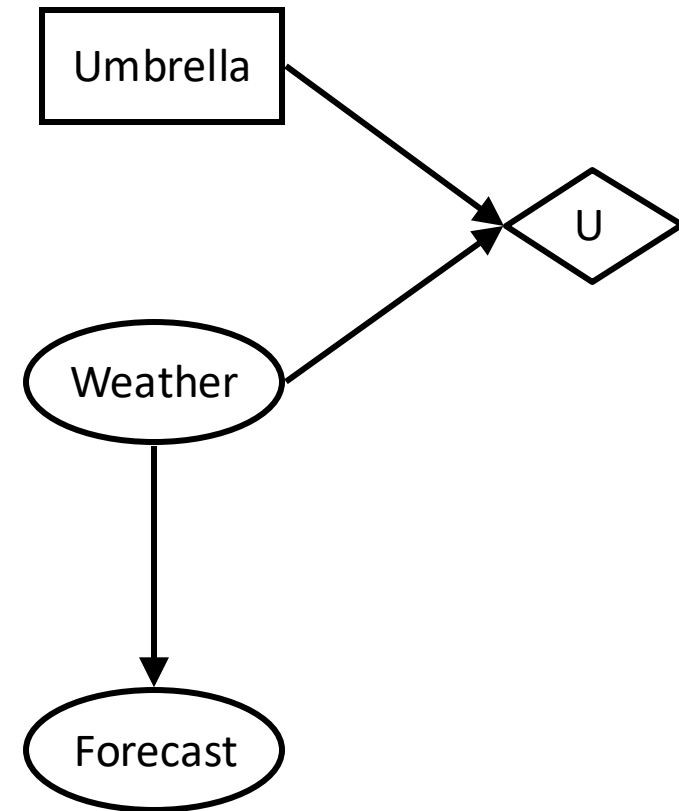


- Utility node (diamond, depends on action and chance nodes)



Decision Networks

- Action selection
 - Instantiate all evidence
 - Set action node(s) each possible way
 - Calculate posterior for all parents of utility node, given the evidence
 - Calculate expected utility for each action
 - Choose maximizing action



Decision Networks

Umbrella = leave

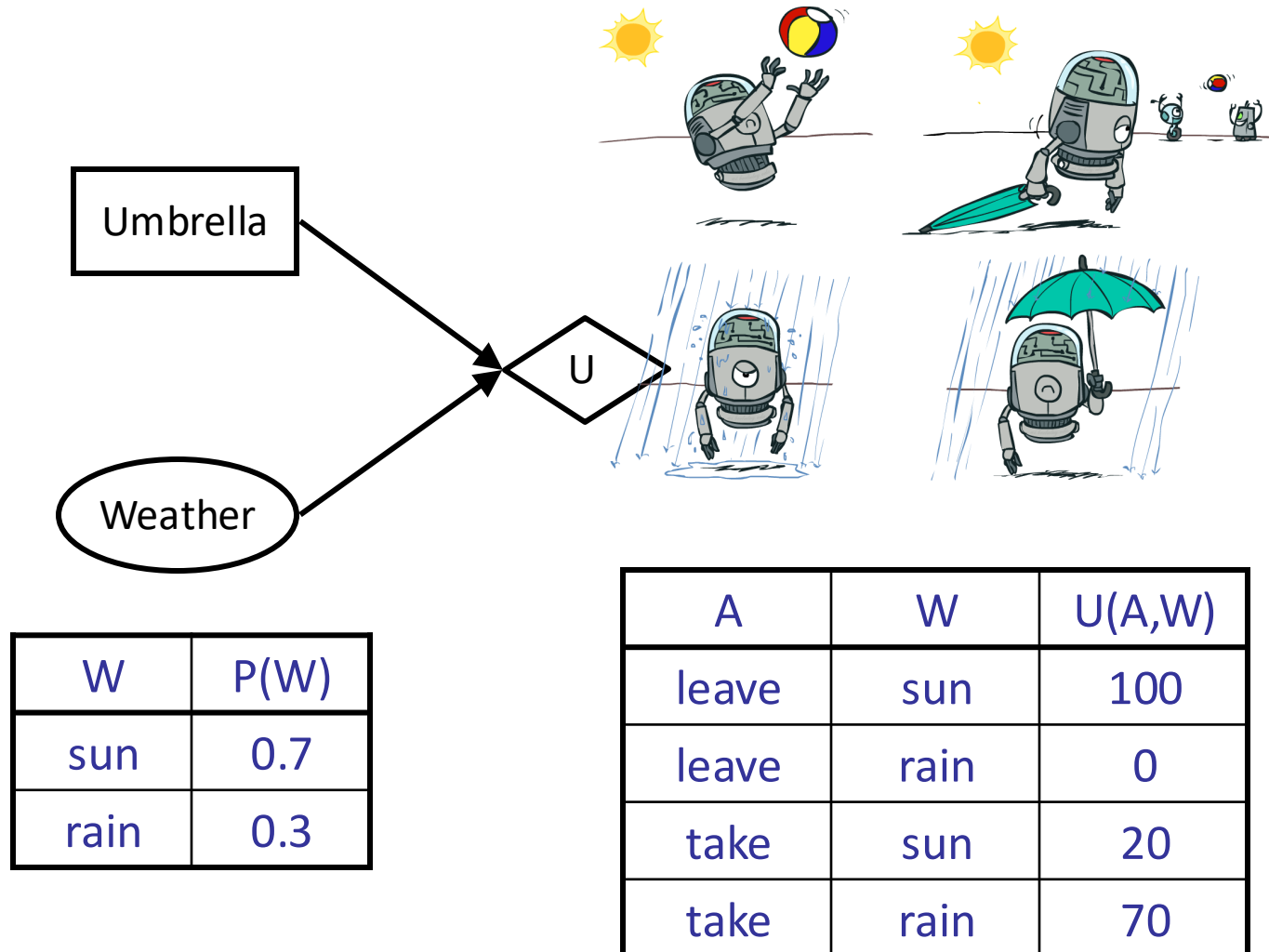
$$\begin{aligned} EU(\text{leave}) &= \sum_w P(w)U(\text{leave}, w) \\ &= 0.7 \cdot 100 + 0.3 \cdot 0 = 70 \end{aligned}$$

Umbrella = take

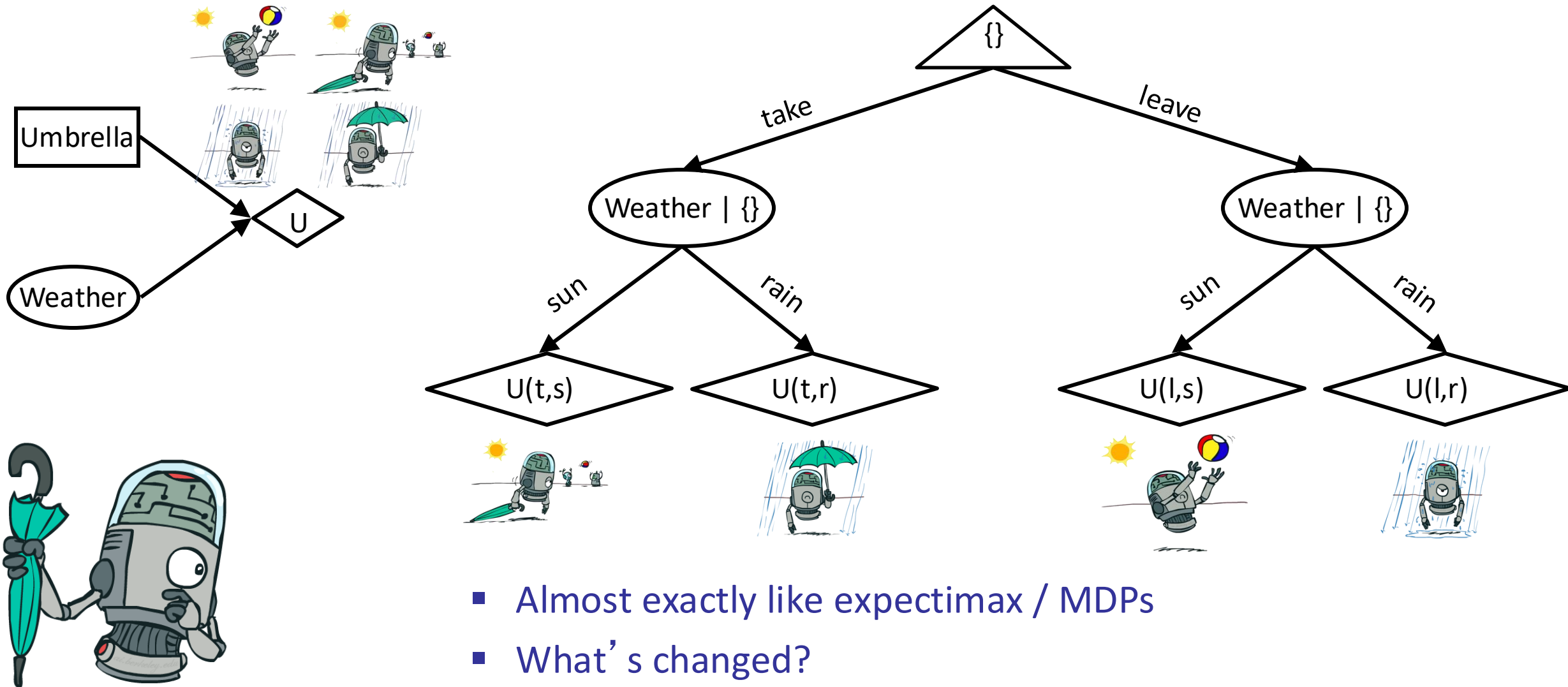
$$\begin{aligned} EU(\text{take}) &= \sum_w P(w)U(\text{take}, w) \\ &= 0.7 \cdot 20 + 0.3 \cdot 70 = 35 \end{aligned}$$

Optimal decision = leave

$$MEU(\emptyset) = \max_a EU(a) = 70$$



Decisions as Outcome Trees



Example: Decision Networks

Umbrella = leave

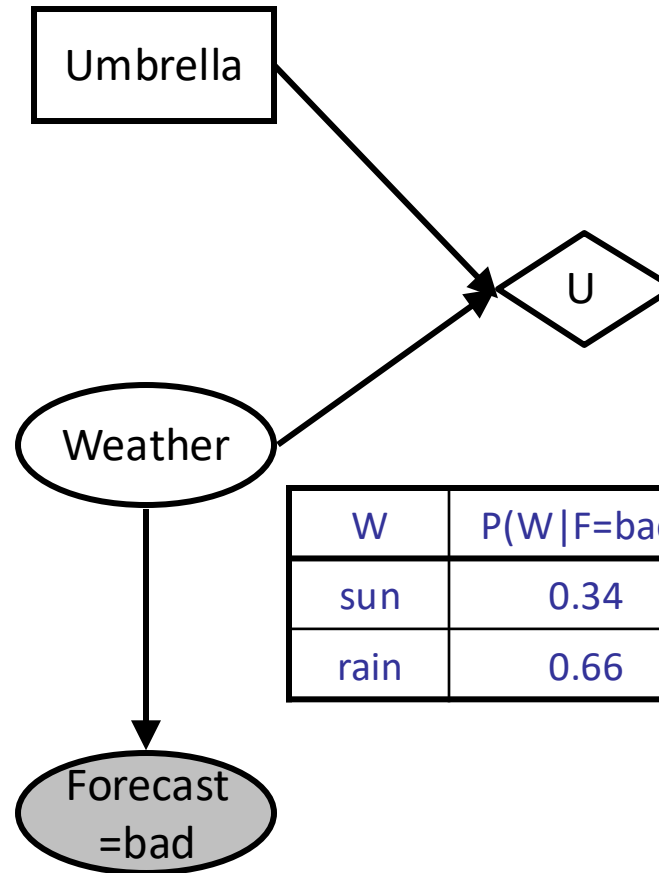
$$\begin{aligned} EU(\text{leave}|\text{bad}) &= \sum_w P(w|\text{bad})U(\text{leave}, w) \\ &= 0.34 \cdot 100 + 0.66 \cdot 0 = 34 \end{aligned}$$

Umbrella = take

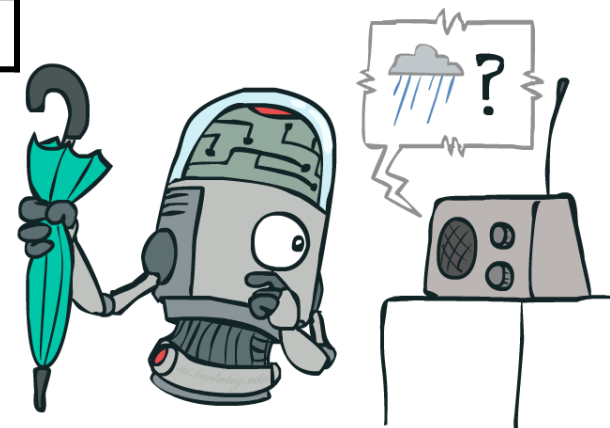
$$\begin{aligned} EU(\text{take}|\text{bad}) &= \sum_w P(w|\text{bad})U(\text{take}, w) \\ &= 0.34 \cdot 20 + 0.66 \cdot 70 = 53 \end{aligned}$$

Optimal decision = take

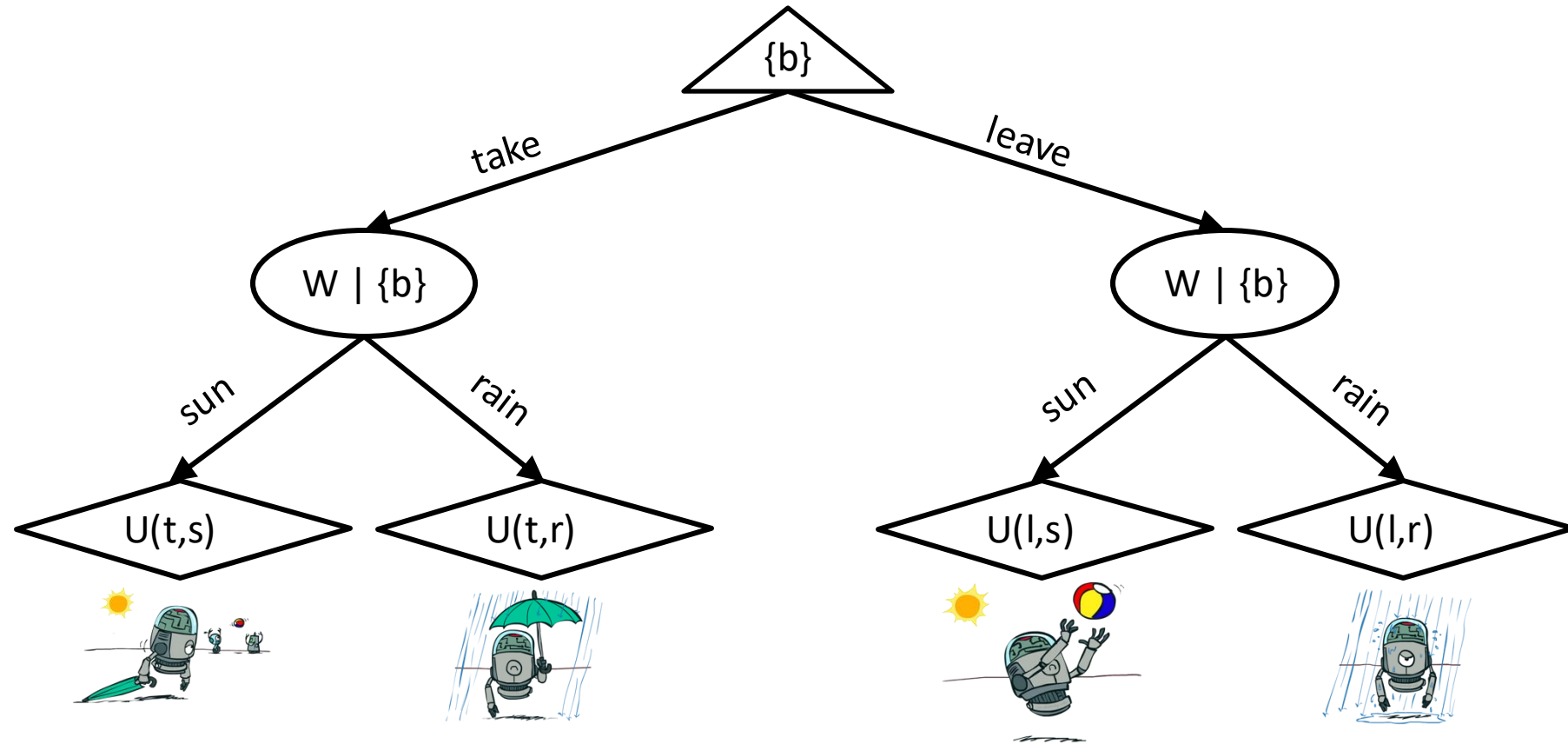
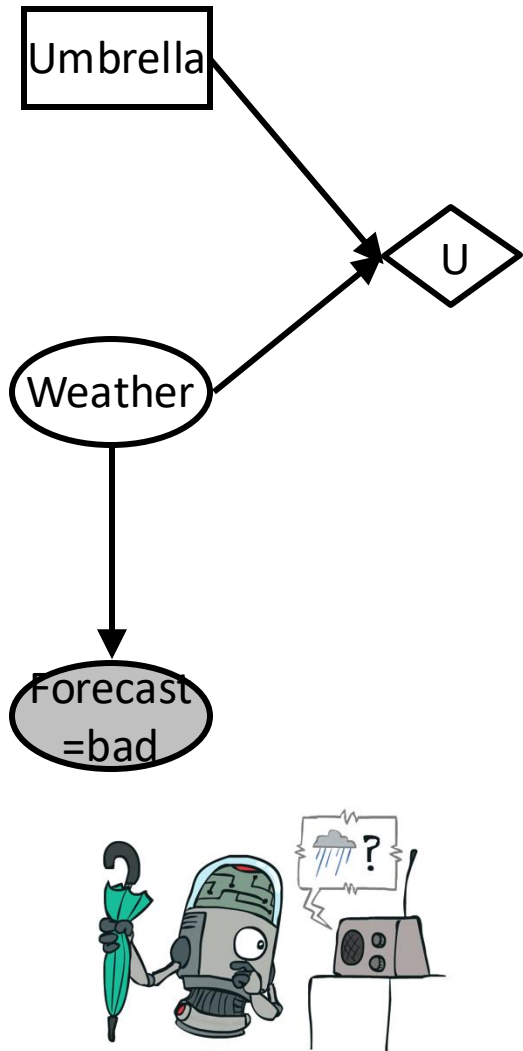
$$MEU(F = \text{bad}) = \max_a EU(a|\text{bad}) = 53$$



A	W	U(A,W)
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

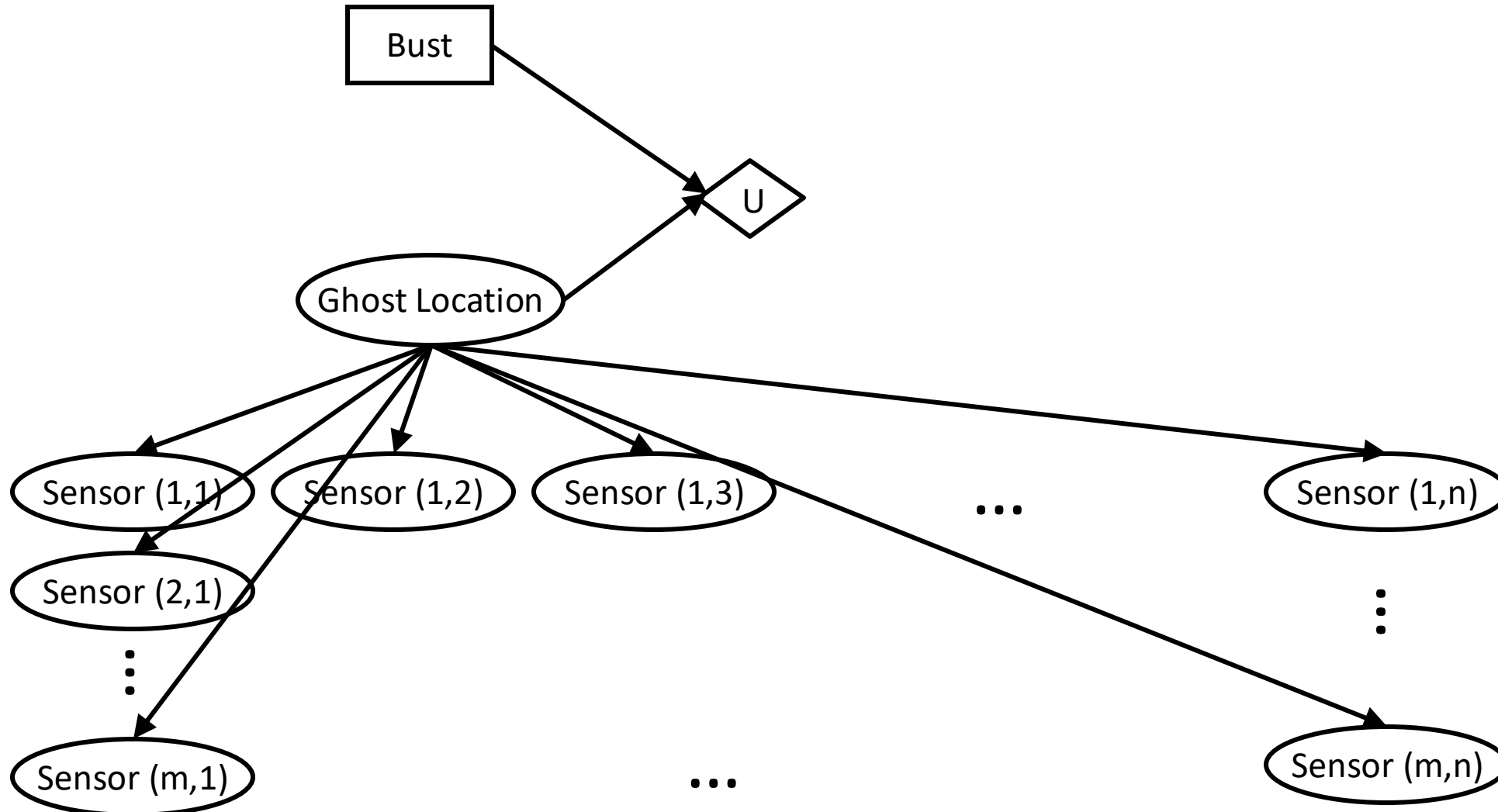


Decisions as Outcome Trees

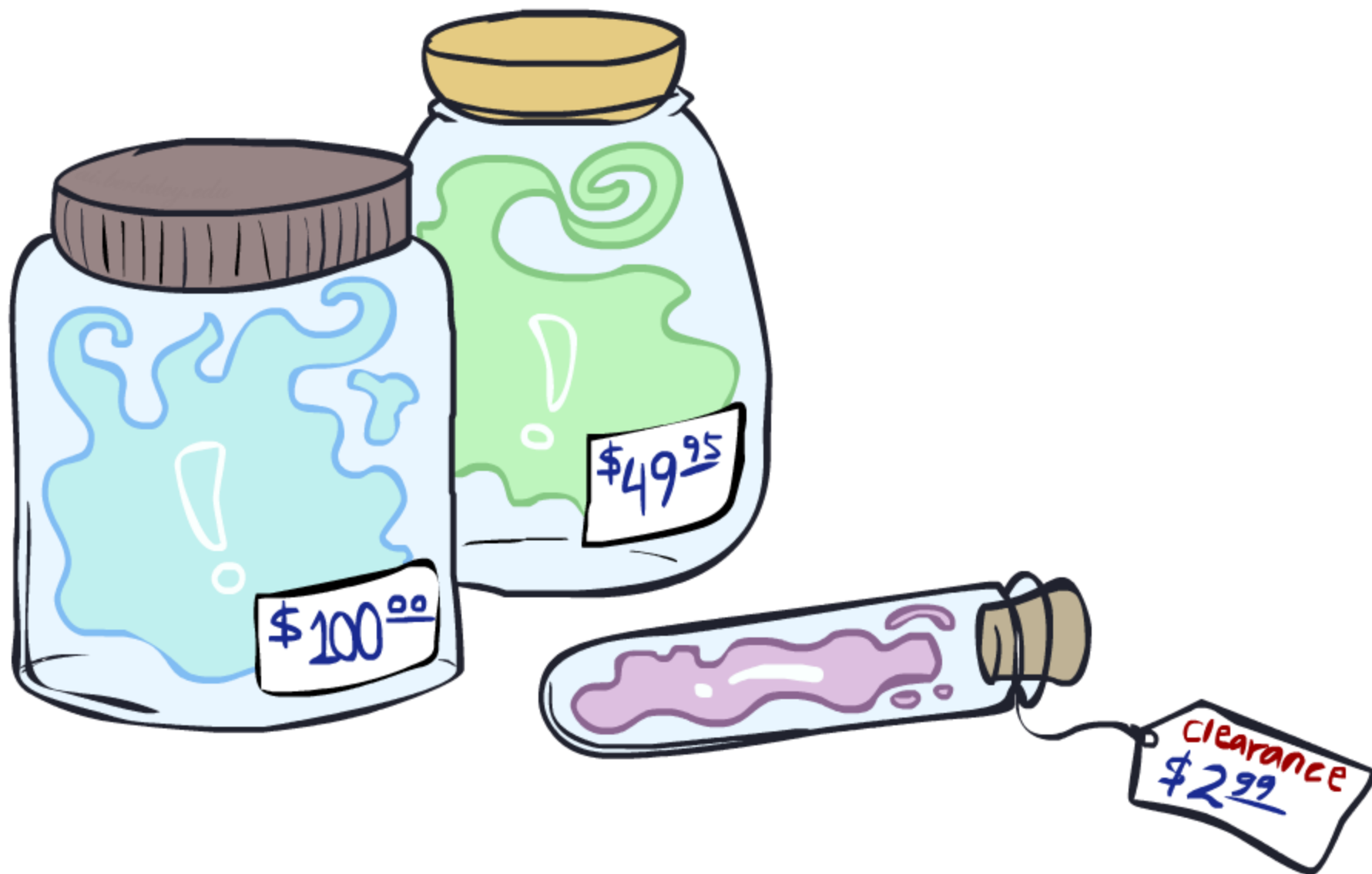


Ghostbusters Decision Network

Demo: Ghostbusters with probability

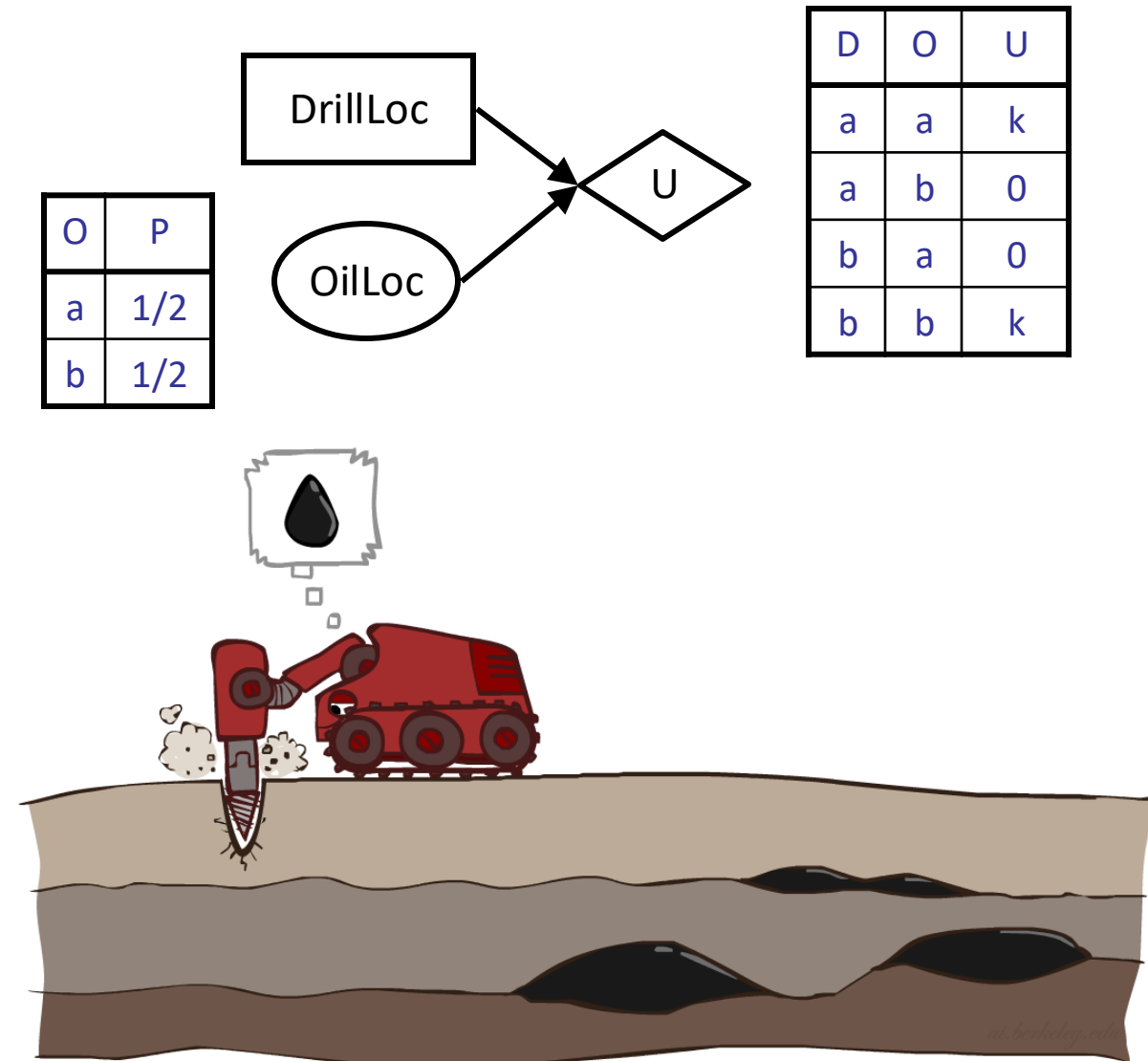


Value of Information



Value of Information

- Idea: compute value of acquiring evidence
 - Can be done directly from decision network
- Example: buying oil drilling rights
 - Two blocks A and B, exactly one has oil, worth k
 - You can drill in one location
 - Prior probabilities 0.5 each, & mutually exclusive
 - Drilling in either A or B has $EU = k/2$, $MEU = k/2$
- Question: what's the value of information of O?
 - Value of knowing which of A or B has oil
 - Value is expected gain in MEU from new info
 - Survey may say "oil in a" or "oil in b," prob 0.5 each
 - If we know OilLoc, MEU is k (either way)
 - Gain in MEU from knowing OilLoc?
 - $VPI(OilLoc) = k/2$
 - Fair price of information: $k/2$



VPI Example: Weather

MEU with no evidence

$$\text{MEU}(\emptyset) = \max_a \text{EU}(a) = 70$$

MEU if forecast is bad

$$\text{MEU}(F = \text{bad}) = \max_a \text{EU}(a|\text{bad}) = 53$$

MEU if forecast is good

$$\text{MEU}(F = \text{good}) = \max_a \text{EU}(a|\text{good}) = 95$$

Forecast distribution

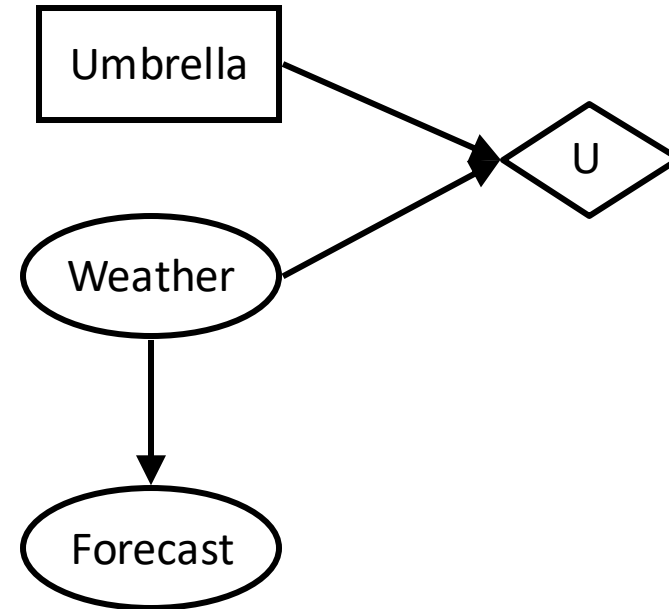
F	P(F)
good	0.59
bad	0.41



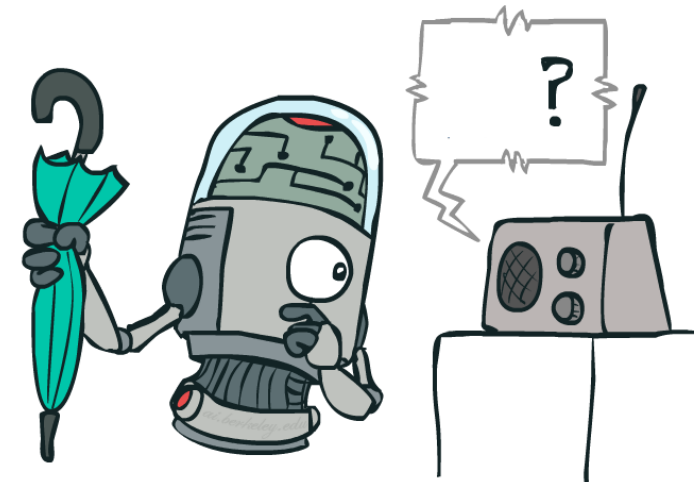
$$0.59 \cdot (95) + 0.41 \cdot (53) - 70$$

$$77.8 - 70 = 7.8$$

$$\text{VPI}(E'|e) = \left(\sum_{e'} P(e'|e) \text{MEU}(e, e') \right) - \text{MEU}(e)$$



A	W	U
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70



Value of Information

- Assume we have evidence $E=e$. Value if we act now:

$$MEU(e) = \max_a \sum_s P(s|e) U(s, a)$$

- Assume we see that $E' = e'$. Value if we act then:

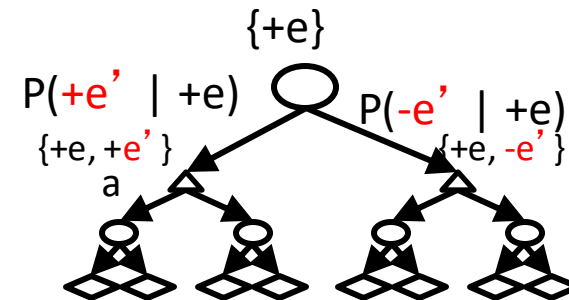
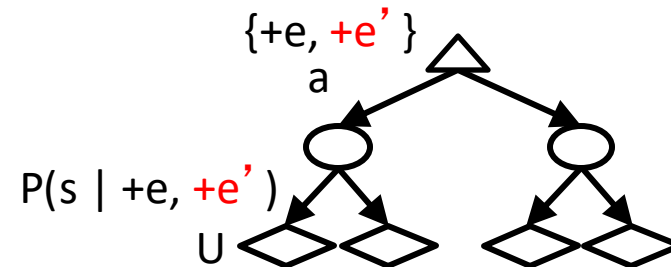
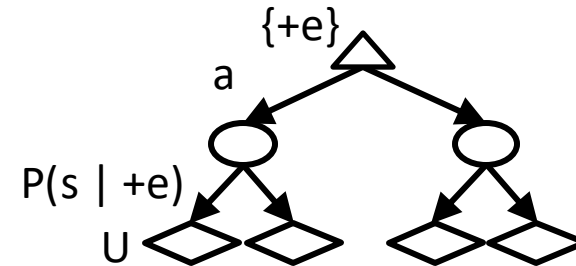
$$MEU(e, e') = \max_a \sum_s P(s|e, e') U(s, a)$$

- BUT E' is a random variable whose value is unknown, so we don't know what e' will be
- Expected value if E' is revealed and then we act:

$$MEU(e, E') = \sum_{e'} P(e'|e) MEU(e, e')$$

- Value of information: how much MEU goes up by revealing E' first then acting, over acting now:

$$VPI(E'|e) = MEU(e, E') - MEU(e)$$



VPI Properties

- Nonnegative

$$\forall E', e : \text{VPI}(E'|e) \geq 0$$



- Nonadditive

(think of observing E_j twice)

$$\text{VPI}(E_j, E_k|e) \neq \text{VPI}(E_j|e) + \text{VPI}(E_k|e)$$



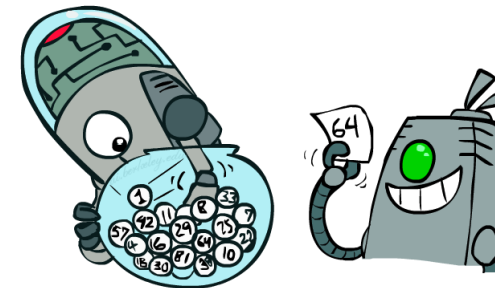
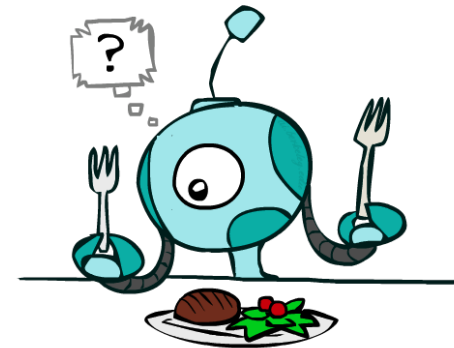
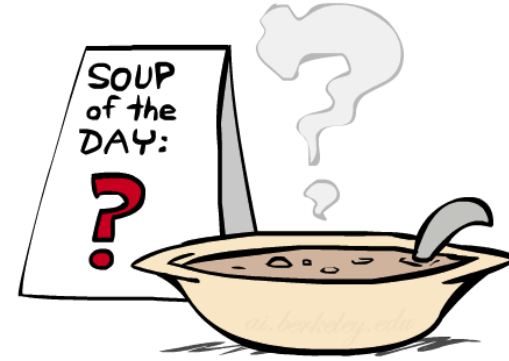
- Order-independent

$$\begin{aligned} \text{VPI}(E_j, E_k|e) &= \text{VPI}(E_j|e) + \text{VPI}(E_k|e, E_j) \\ &= \text{VPI}(E_k|e) + \text{VPI}(E_j|e, E_k) \end{aligned}$$



Quick VPI Questions

- The soup of the day is either clam chowder or split pea, but you wouldn't order either one. What's the value of knowing which it is?
- There are two kinds of plastic forks at a picnic. One kind is slightly sturdier. What's the value of knowing which?
- You're playing the lottery. The prize will be \$0 or \$100. You can play any number between 1 and 100 (chance of winning is 1%). What is the value of knowing the winning number?



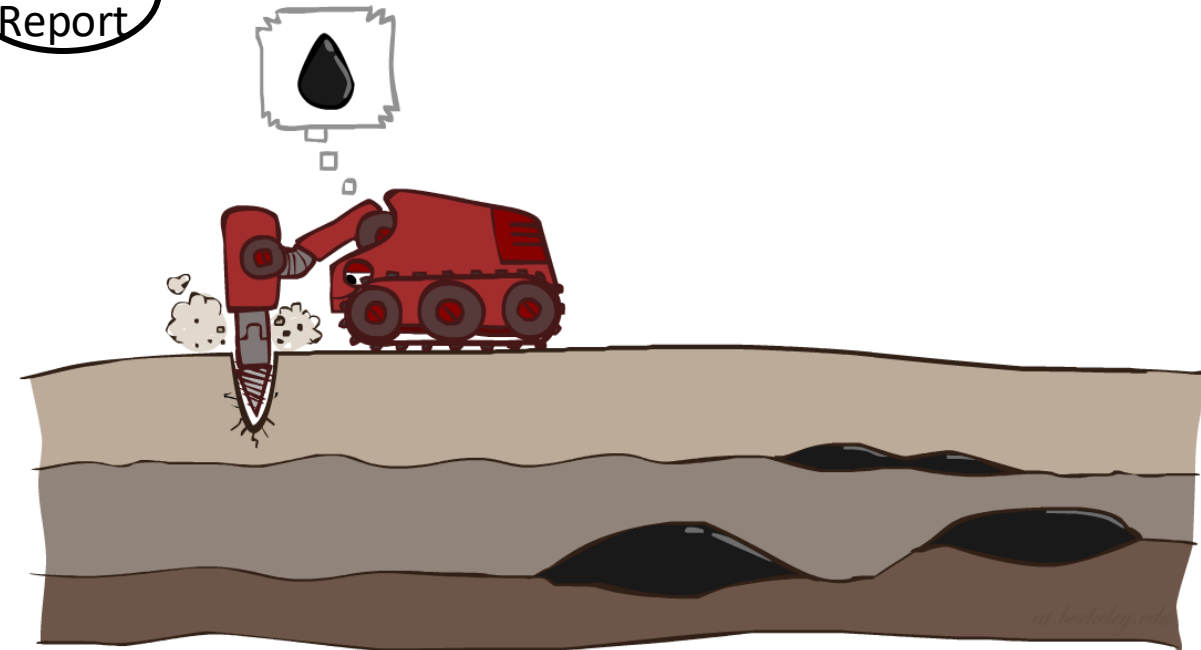
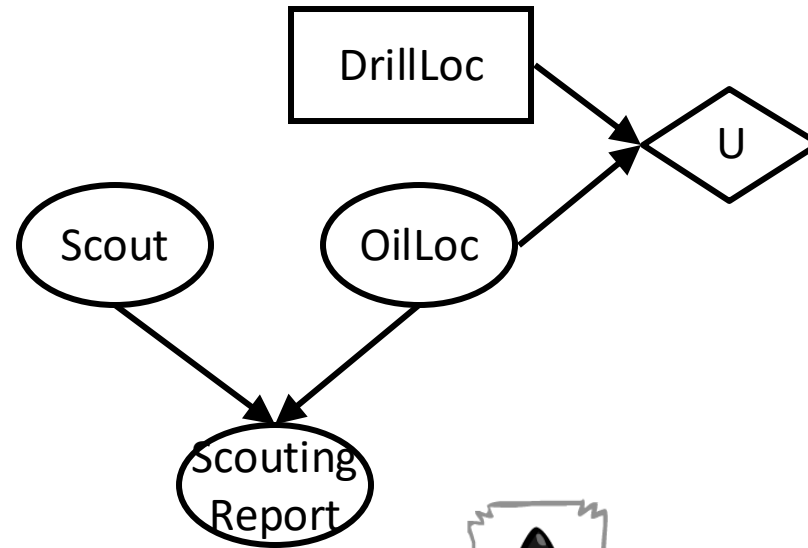
Value of Imperfect Information?



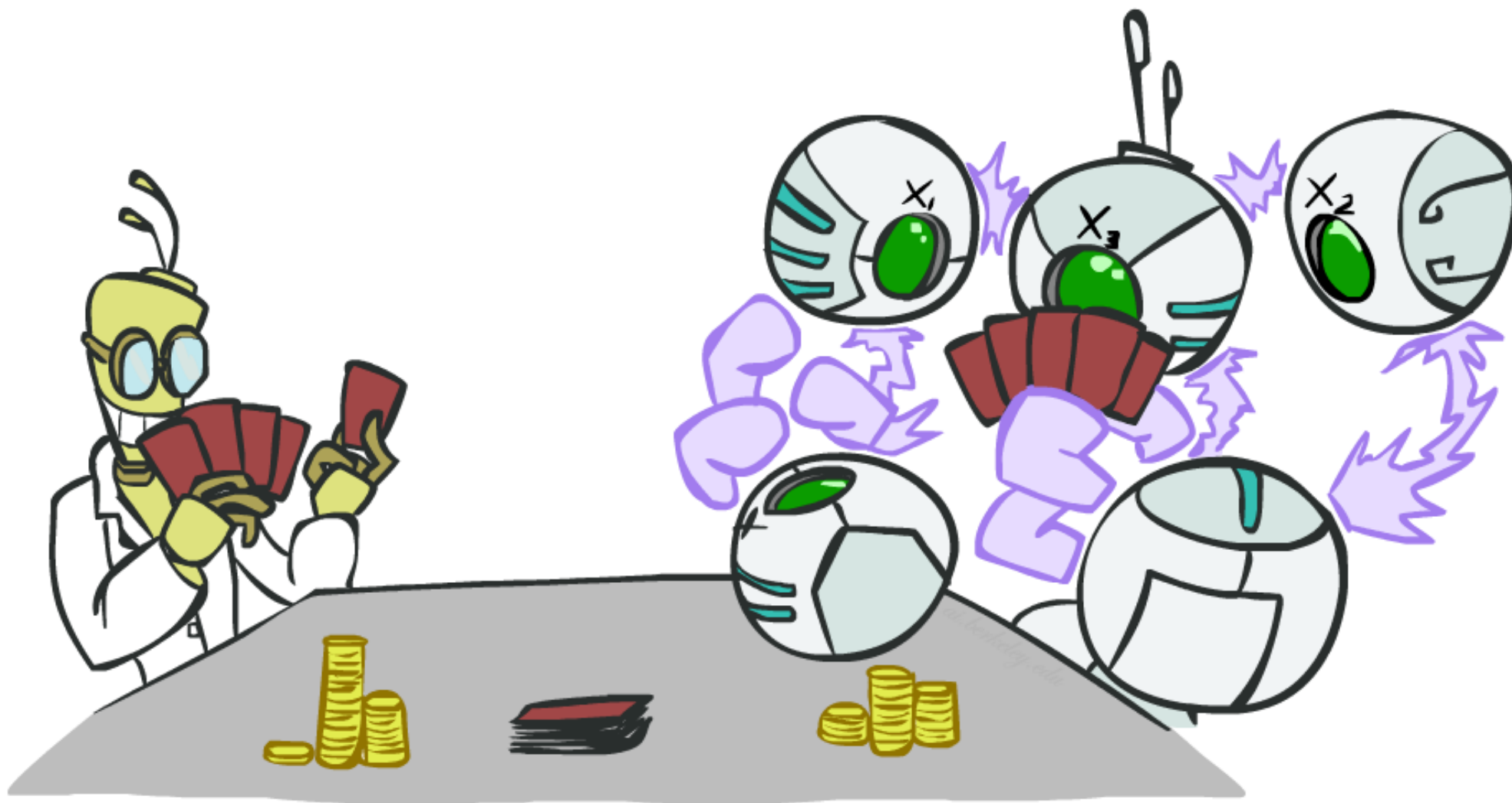
- No such thing
- Information corresponds to the observation of a node in the decision network
- If data is “noisy” that just means we don’t observe the original variable, but another variable which is a noisy version of the original one

VPI Question

- $VPI(\text{OilLoc})$?
- $VPI(\text{ScoutingReport})$?
- $VPI(\text{Scout})$?
- $VPI(\text{Scout} \mid \text{ScoutingReport})$?
- Generally:
If $\text{Parents}(U) \perp\!\!\!\perp Z \mid \text{CurrentEvidence}$
Then $VPI(Z \mid \text{CurrentEvidence}) = 0$

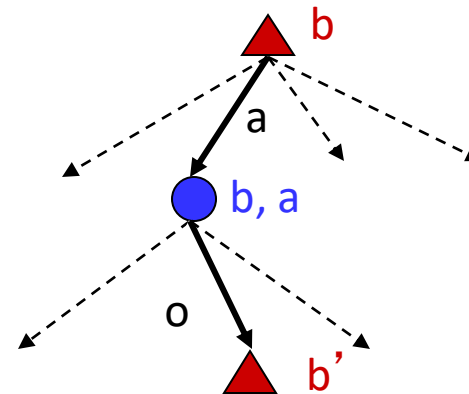
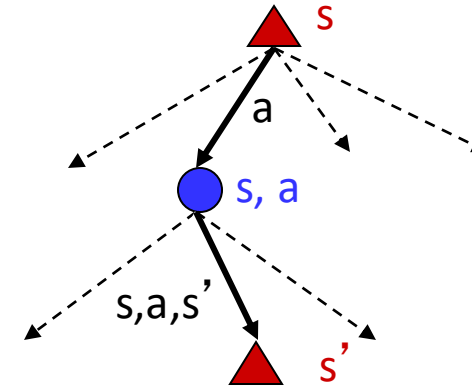


POMDPs



POMDPs

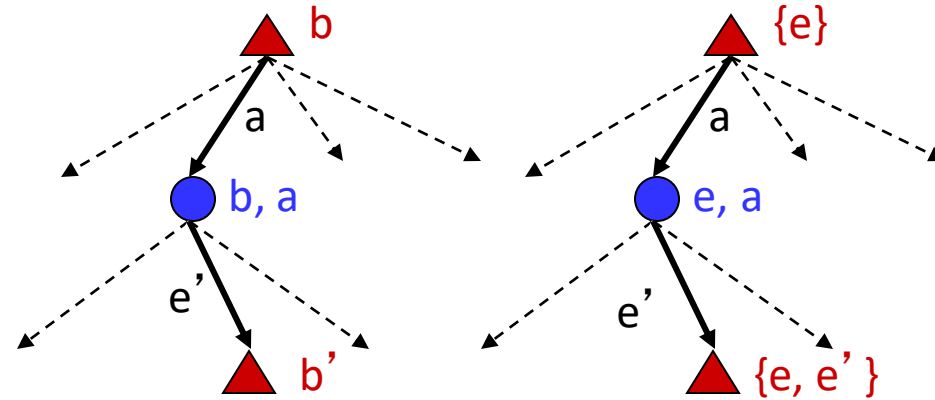
- MDPs have:
 - States S
 - Actions A
 - Transition function $P(s' | s, a)$ (or $T(s, a, s')$)
 - Rewards $R(s, a, s')$
- POMDPs add:
 - Observations O
 - Observation function $P(o | s)$ (or $O(s, o)$)
- POMDPs are MDPs over belief states b (distributions over S)
- We'll be able to say more in a few lectures



Example: Ghostbusters

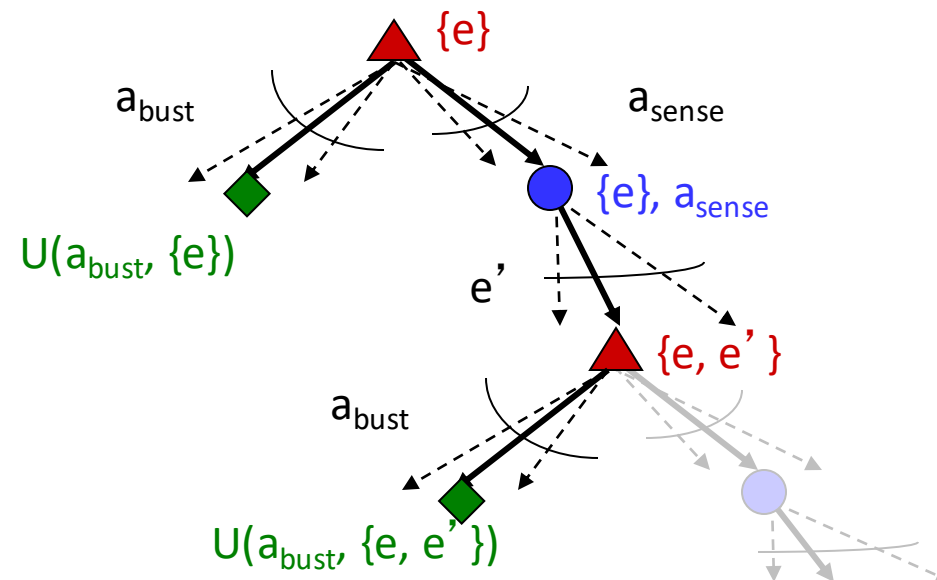
■ In (static) Ghostbusters:

- Belief state determined by evidence to date $\{e\}$
- Tree really over evidence sets
- Probabilistic reasoning needed to predict new evidence given past evidence



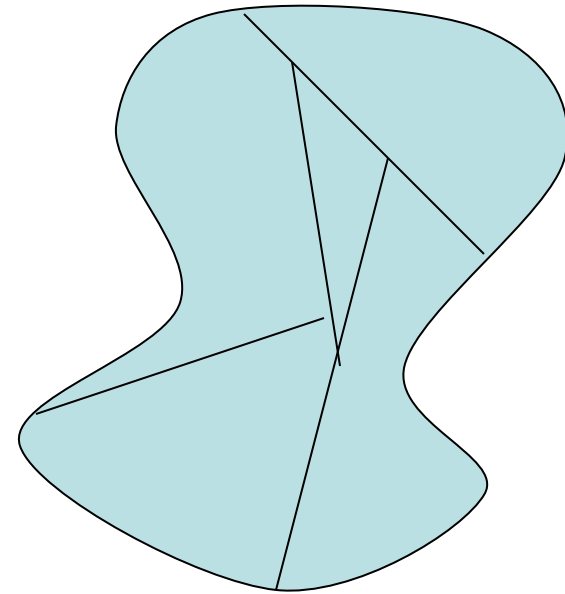
■ Solving POMDPs

- One way: use truncated expectimax to compute approximate value of actions
- What if you only considered busting or one sense followed by a bust?
- You get a VPI-based agent!



More Generally*

- General solutions map belief functions to actions
 - Can divide regions of belief space (set of belief functions) into policy regions (gets complex quickly)
 - Can build approximate policies using discretization methods
 - Can factor belief functions in various ways
- Overall, POMDPs are very (actually PSACE-) hard
- Most real problems are POMDPs, but we can rarely solve them in general!



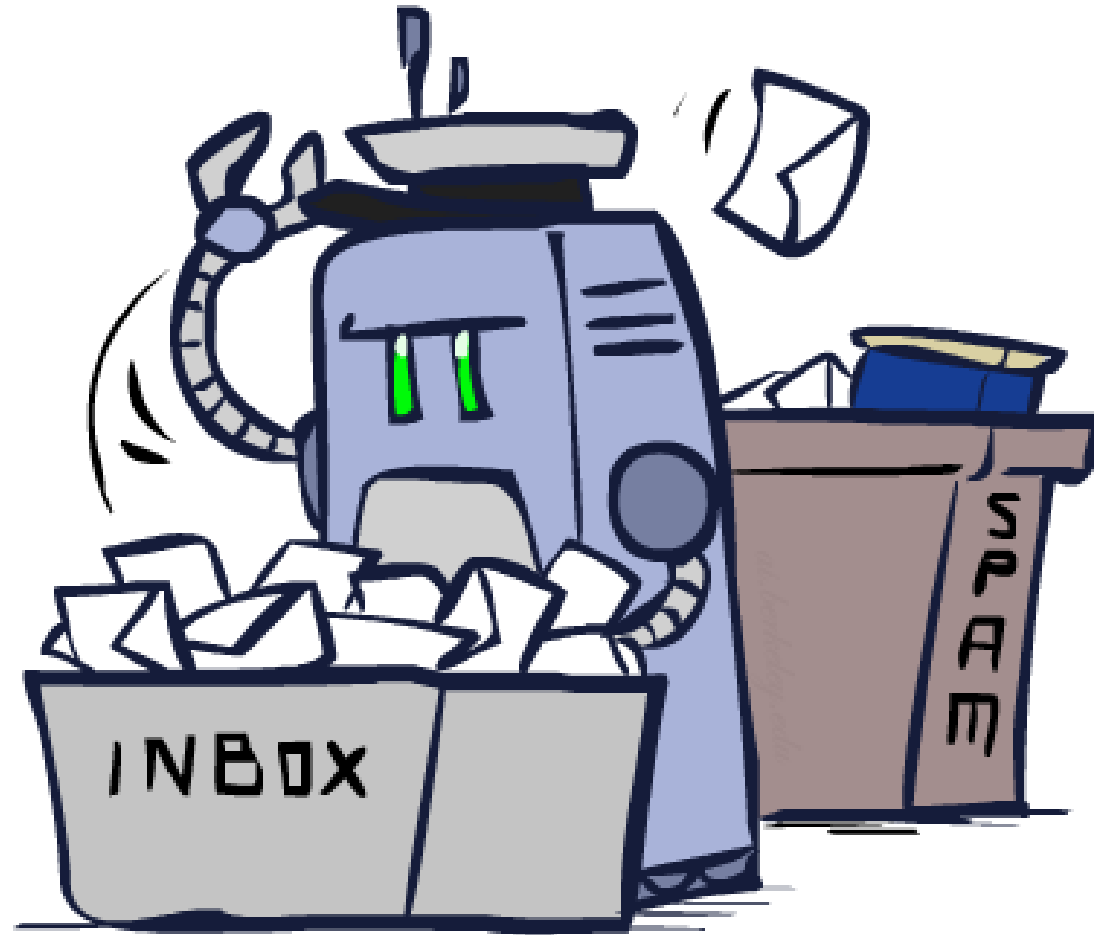
Naïve Bayes



Machine Learning

- Up until now: how use a model to make optimal decisions
- Machine learning: how to acquire a model from data / experience
 - Learning parameters (e.g. probabilities)
 - Learning structure (e.g. BN graphs)
 - Learning hidden concepts (e.g. clustering)
- Today: model-based classification with Naive Bayes

Classification



Example: Spam Filter

- Input: an email
- Output: spam/ham
- Setup:
 - Get a large collection of example emails, each labeled “spam” or “ham”
 - Note: someone has to hand label all this data!
 - Want to learn to predict labels of new, future emails
- Features: The attributes used to make the ham / spam decision
 - Words: FREE!
 - Text Patterns: \$dd, CAPS
 - Non-text: SenderInContacts
 - ...



Dear Sir.

First, I must solicit your confidence in this transaction, this is by virtue of its nature as being utterly confidential and top secret. ...



TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

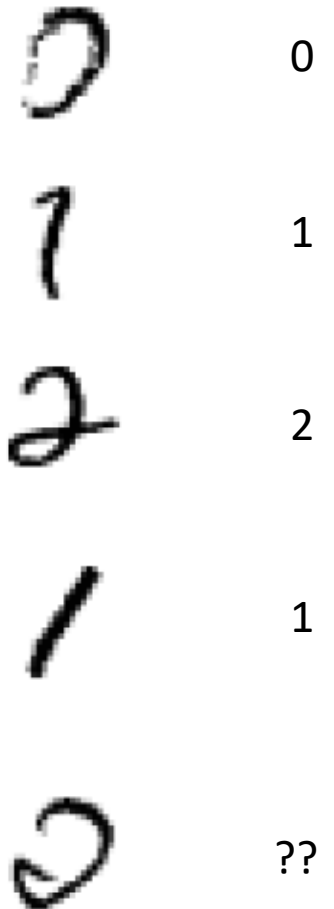
99 MILLION EMAIL ADDRESSES FOR ONLY \$99



Ok, I know this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

Example: Digit Recognition

- Input: images / pixel grids
- Output: a digit 0-9
- Setup:
 - Get a large collection of example images, each labeled with a digit
 - Note: someone has to hand label all this data!
 - Want to learn to predict labels of new, future digit images
- Features: The attributes used to make the digit decision
 - Pixels: (6,8)=ON
 - Shape Patterns: NumComponents, AspectRatio, NumLoops
 - ...

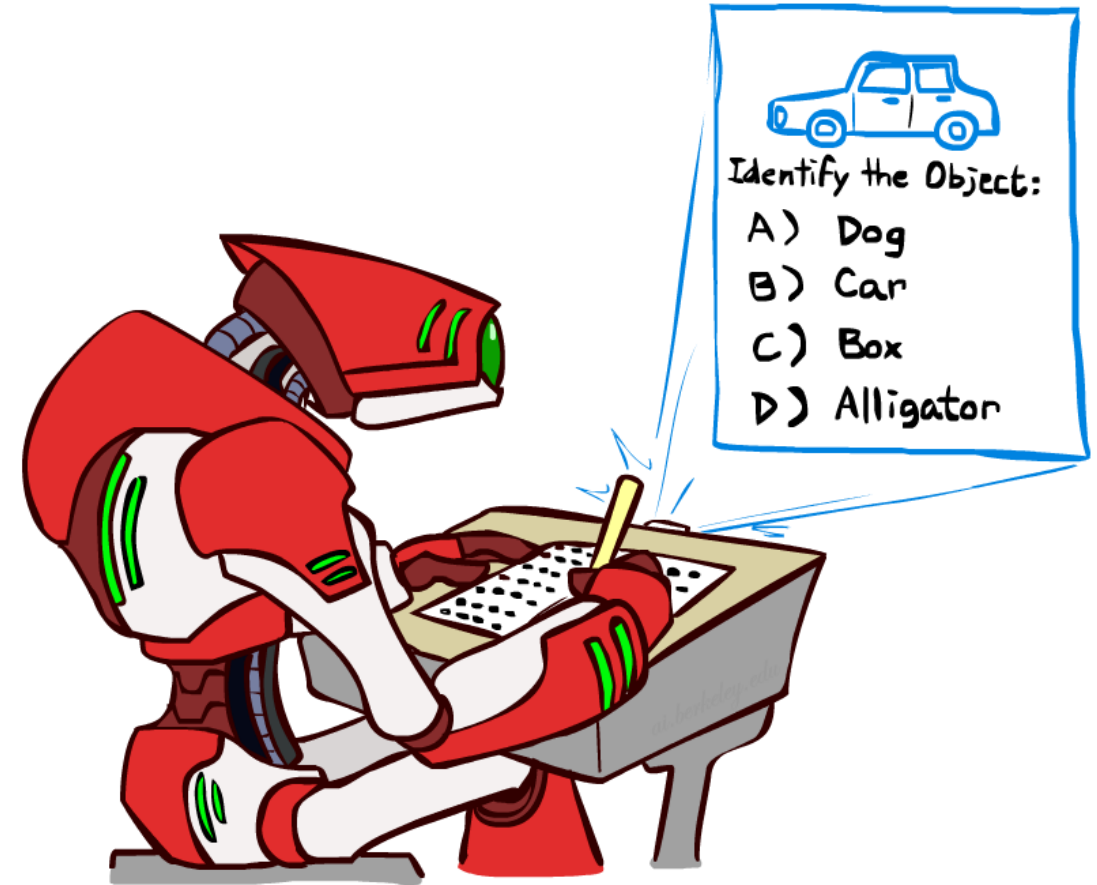


Other Classification Tasks

- Classification: given inputs x , predict labels (c)

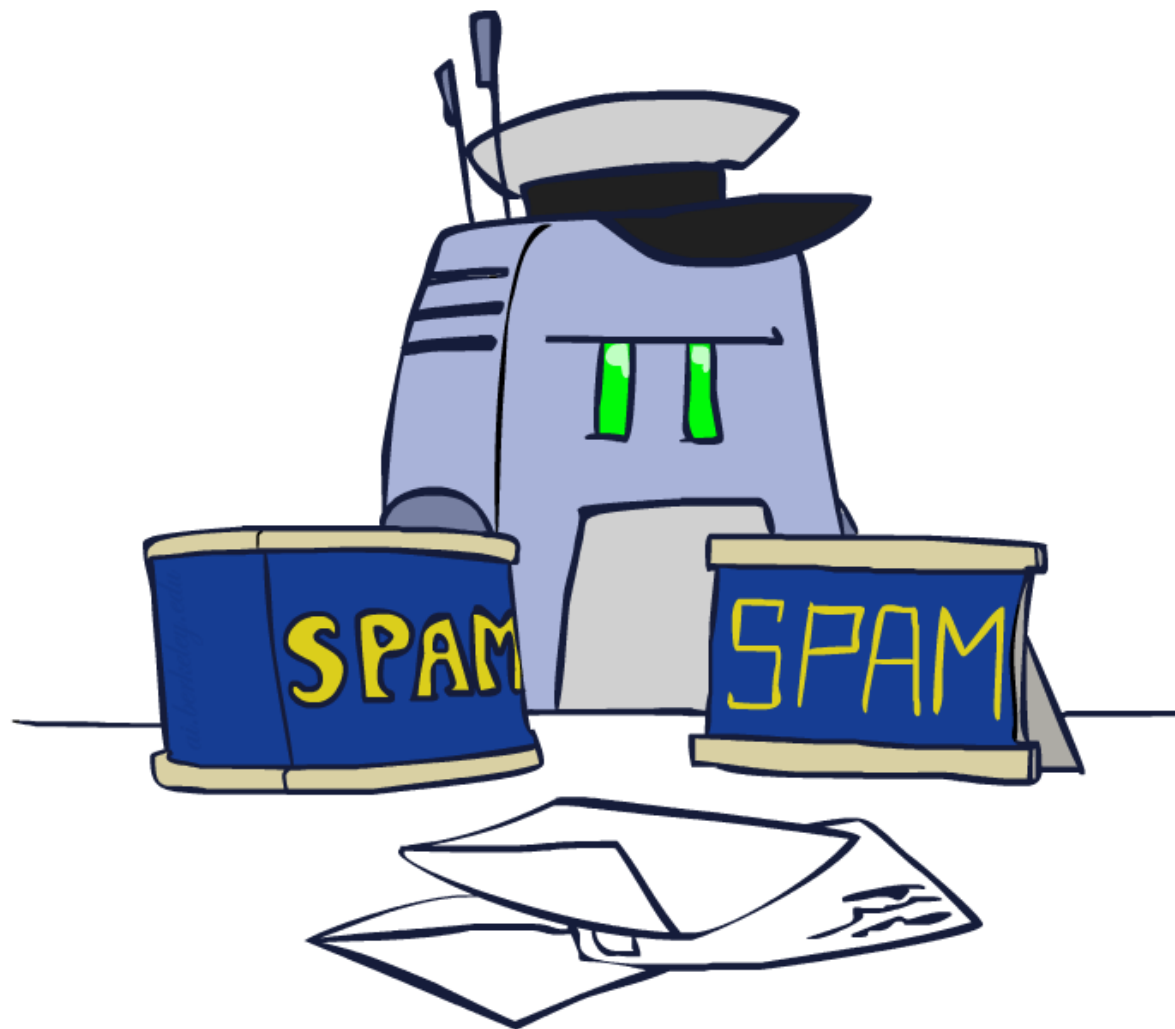
- Examples:

- Spam detection (input: document, classes: spam / ham)
- OCR (input: images, classes: characters)
- Medical diagnosis (input: symptoms, classes: diseases)
- Automatic essay grading (input: document, classes: grades)
- Fraud detection (input: account activity, classes: fraud / no fraud)
- Customer service email routing
- ... many more



- Classification is an important commercial technology!

Model-Based Classification



Model-Based Classification

- Model-based approach
 - Build a model (e.g. Bayes' net) where both the label and features are random variables
 - Instantiate any observed features
 - Query for the distribution of the label conditioned on the features
- Challenges
 - What structure should the BN have?
 - How should we learn its parameters?




Naïve Bayes for Digits

- Naïve Bayes: Assume all features are independent effects of the label

- Simple digit recognition version:

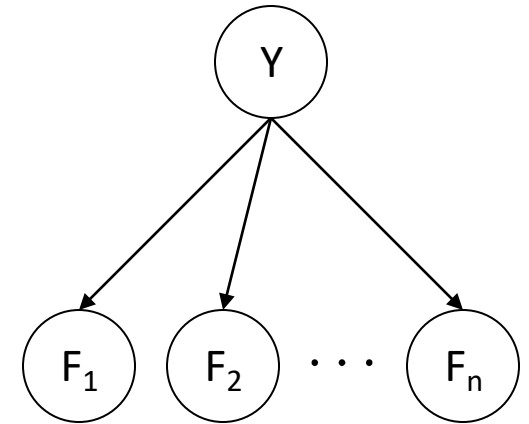
- One feature (variable) F_{ij} for each grid position $\langle i,j \rangle$
- Feature values are on / off, based on whether intensity is more or less than 0.5 in underlying image
- Each input maps to a feature vector, e.g.

 $\rightarrow \langle F_{0,0} = 0 \ F_{0,1} = 0 \ F_{0,2} = 1 \ F_{0,3} = 1 \ F_{0,4} = 0 \ \dots F_{15,15} = 0 \rangle$

- Here: lots of features, each is binary valued

- Naïve Bayes model: $P(Y|F_{0,0} \dots F_{15,15}) \propto P(Y) \prod_{i,j} P(F_{i,j}|Y)$

- What do we need to learn?



General Naïve Bayes

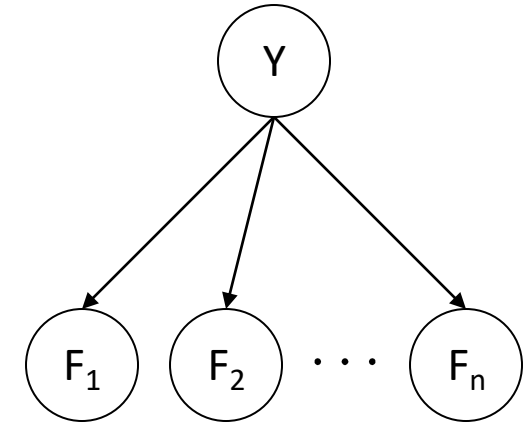
- A general Naive Bayes model:

$|Y|$ parameters

$$P(Y, F_1 \dots F_n) = P(Y) \prod_i P(F_i|Y)$$

$|Y| \times |F|^n$ values

$n \times |F| \times |Y|$
parameters



- We only have to specify how each feature depends on the class
- Total number of parameters is *linear* in n
- Model is very simplistic, but often works anyway

Inference for Naïve Bayes

- Goal: compute posterior distribution over label variable Y
 - Step 1: get joint probability of label and evidence for each label

$$P(Y, f_1 \dots f_n) = \begin{bmatrix} P(y_1, f_1 \dots f_n) \\ P(y_2, f_1 \dots f_n) \\ \vdots \\ P(y_k, f_1 \dots f_n) \end{bmatrix} \Rightarrow \begin{bmatrix} P(y_1) \prod_i P(f_i|y_1) \\ P(y_2) \prod_i P(f_i|y_2) \\ \vdots \\ P(y_k) \prod_i P(f_i|y_k) \end{bmatrix}$$

$$P(f_1 \dots f_n)$$

+ ↶

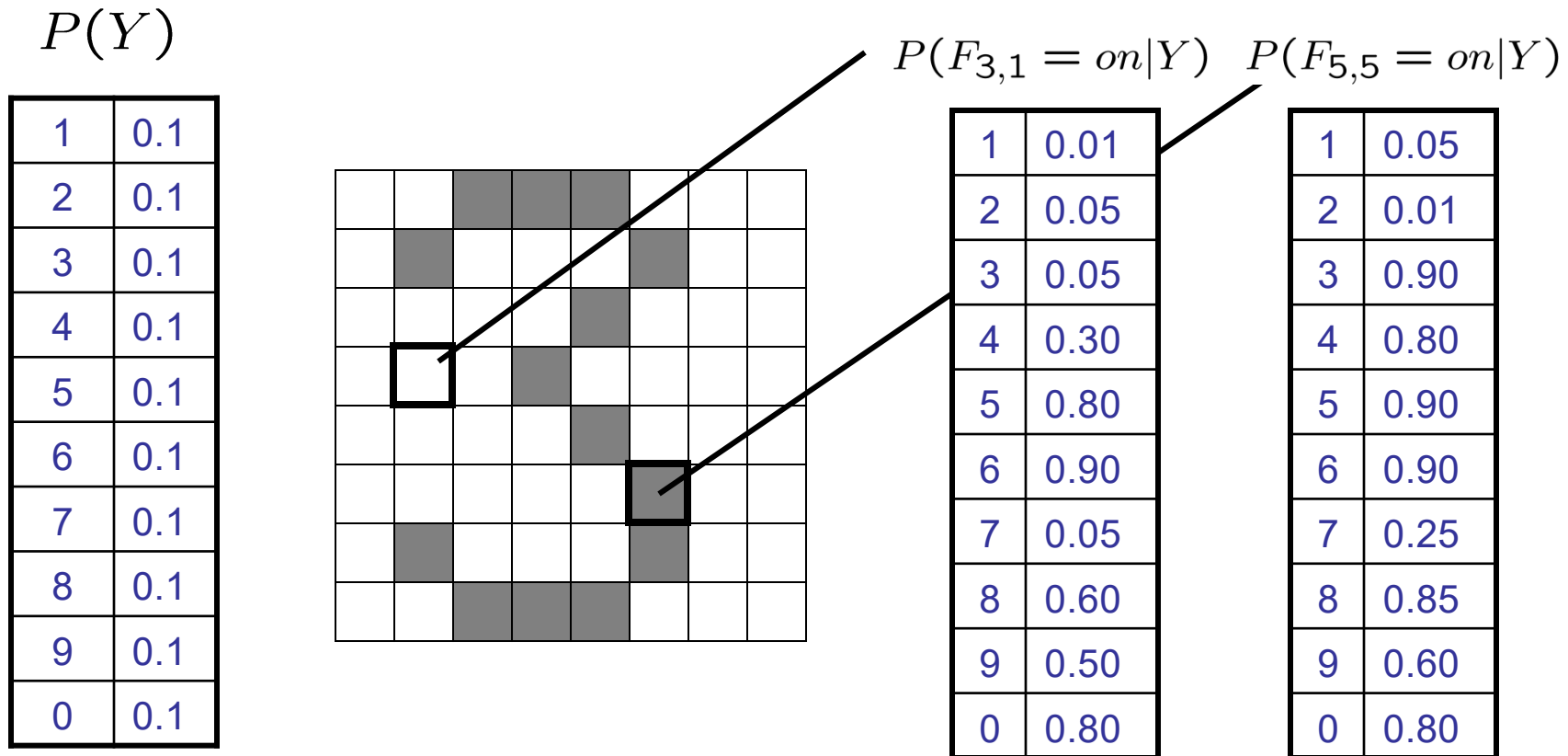
- Step 2: sum to get probability of evidence
- Step 3: normalize by dividing Step 1 by Step 2

$$P(Y|f_1 \dots f_n)$$

General Naïve Bayes

- What do we need in order to use Naïve Bayes?
 - Inference method (we just saw this part)
 - Start with a bunch of probabilities: $P(Y)$ and the $P(F_i|Y)$ tables
 - Use standard inference to compute $P(Y|F_1...F_n)$
 - Nothing new here
 - Estimates of local conditional probability tables
 - $P(Y)$, the prior over labels
 - $P(F_i|Y)$ for each feature (evidence variable)
 - These probabilities are collectively called the *parameters* of the model and denoted by θ
 - Up until now, we assumed these appeared by magic, but...
 - ...they typically come from training data counts: we'll look at this soon

Example: Conditional Probabilities



Naïve Bayes for Text

- Bag-of-words Naïve Bayes:

- Features: W_i is the word at position i
- As before: predict label conditioned on feature variables (spam vs. ham)
- As before: assume features are conditionally independent given label
- New: each W_i is identically distributed

- Generative model: $P(Y, W_1 \dots W_n) = P(Y) \prod_i P(W_i|Y)$

*Word at position
 i , not i^{th} word in
the dictionary!*

- “Tied” distributions and bag-of-words

- Usually, each variable gets its own conditional probability distribution $P(F|Y)$
- In a bag-of-words model
 - Each position is identically distributed
 - All positions share the same conditional probs $P(W|Y)$
 - Why make this assumption?
- Called “bag-of-words” because model is insensitive to word order or reordering

Example: Spam Filtering

- Model: $P(Y, W_1 \dots W_n) = P(Y) \prod_i P(W_i|Y)$
- What are the parameters?

$P(Y)$

ham : 0.66
spam: 0.33

$P(W|\text{spam})$

the : 0.0156
to : 0.0153
and : 0.0115
of : 0.0095
you : 0.0093
a : 0.0086
with: 0.0080
from: 0.0075
...

$P(W|\text{ham})$

the : 0.0210
to : 0.0133
of : 0.0119
2002: 0.0110
with: 0.0108
from: 0.0107
and : 0.0105
a : 0.0100
...

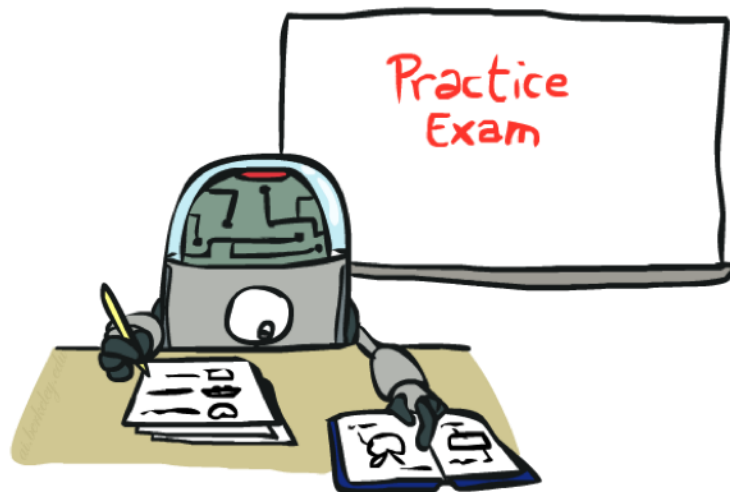
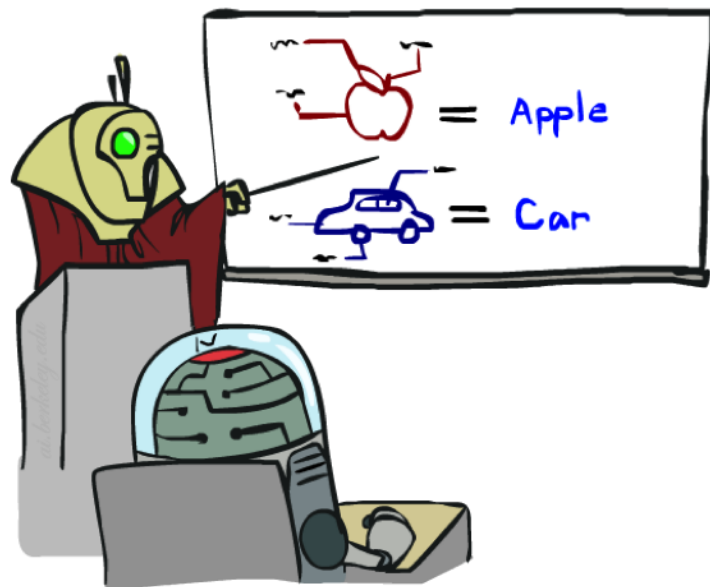
- Where do these tables come from?

Spam Example

Word	$P(w \text{spam})$	$P(w \text{ham})$	Tot Spam	Tot Ham
(prior)	0.33333	0.66666	-1.1	-0.4

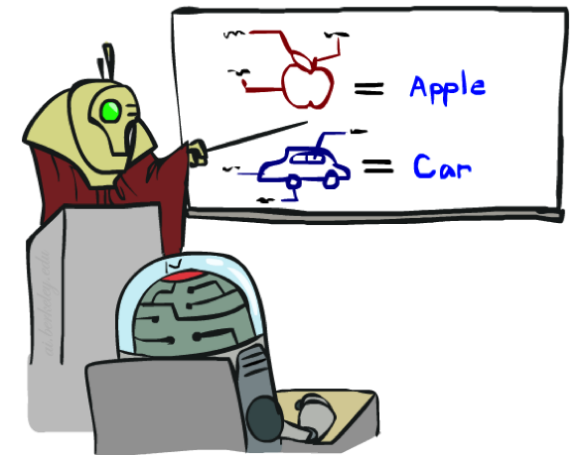
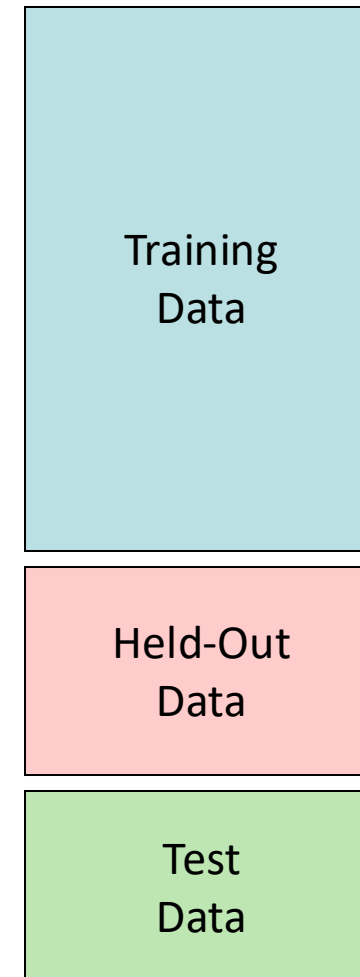
$$P(\text{spam} | w) = 98.9$$

Training and Testing



Important Concepts

- Data: labeled instances, e.g. emails marked spam/ham
 - Training set
 - Held out set
 - Test set
- Features: attribute-value pairs which characterize each x
- Experimentation cycle
 - Learn parameters (e.g. model probabilities) on training set
 - (Tune hyperparameters on held-out set)
 - Compute accuracy of test set
 - Very important: never “peek” at the test set!
- Evaluation
 - Accuracy: fraction of instances predicted correctly
- Overfitting and generalization
 - Want a classifier which does well on *test* data
 - Overfitting: fitting the training data very closely, but not generalizing well
 - We’ll investigate overfitting and generalization formally in a few lectures

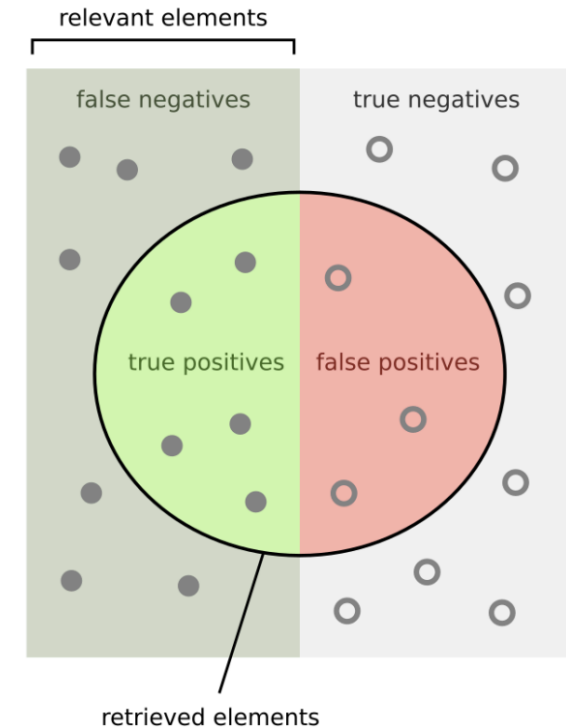


Confusion Matrix

- Used to show space of actual and predicted values

		Predicted	
		Spam	Not spam
Actual	Spam	True positive ✓	False negative ✗
	Not spam	False positive ✗	True negative ✓

From evidentlyai.com



How many retrieved items are relevant?

Precision =



How many relevant items are retrieved?

Recall =



Performance Metrics

- Accuracy

- Number of correct predictions from the entire data set:
 $(TP + TN) / (TP + FP + TN + FN)$

- Precision

- Number of correct positive predictions from total positive predictions:
 $TP / (TP + FP)$

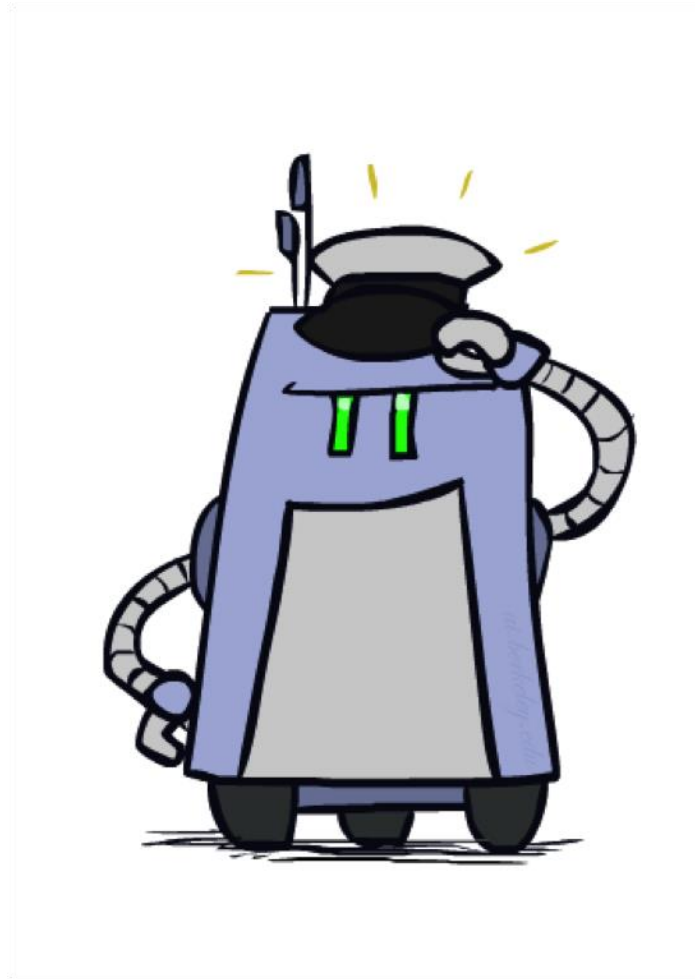
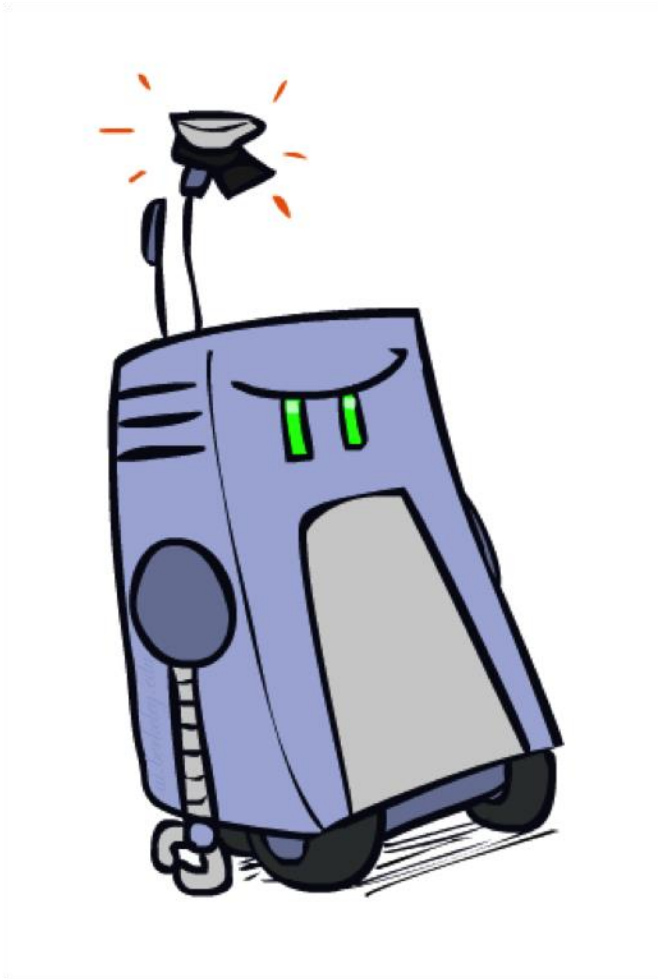
- Recall

- Number of correct positive predictions from the actual positive samples:
 $TP / (TP + FN)$

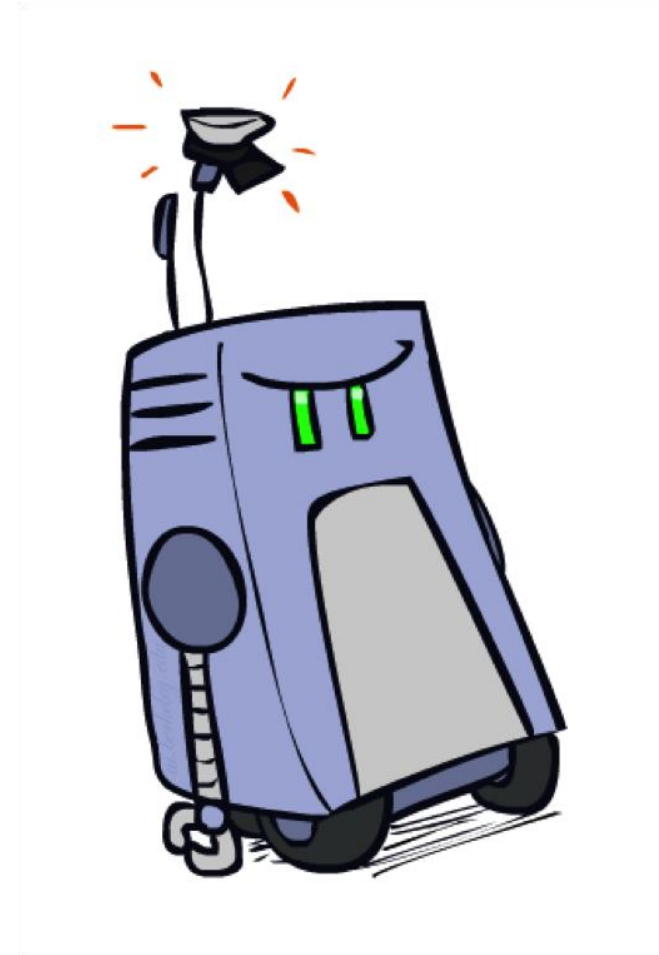
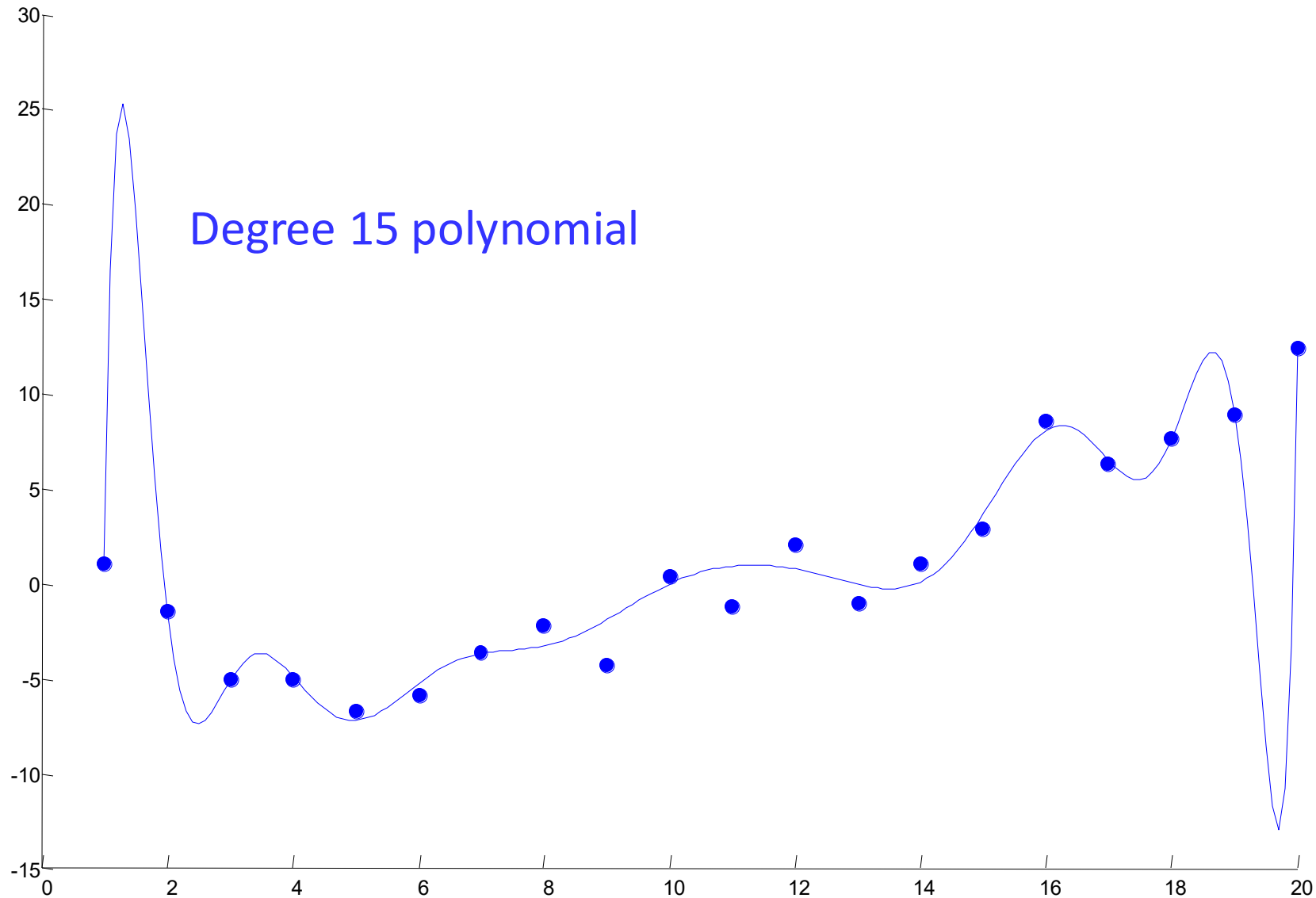
- F-score

- Harmonic mean of Precision and Recall: $2TP / (2TP + FP + FN)$

Generalization and Overfitting



Overfitting



Example: Overfitting

$P(\text{features}, C = 2)$

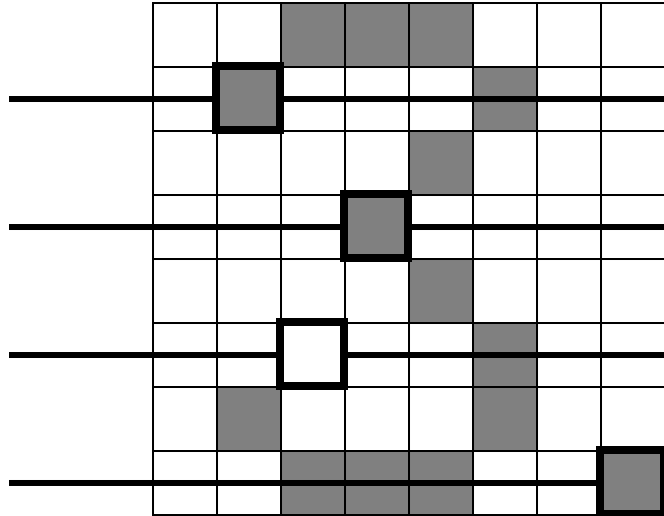
$P(C = 2) = 0.1$

$P(\text{on}|C = 2) = 0.8$

$P(\text{on}|C = 2) = 0.1$

$P(\text{off}|C = 2) = 0.1$

$P(\text{on}|C = 2) = 0.01$



$P(\text{features}, C = 3)$

$P(C = 3) = 0.1$

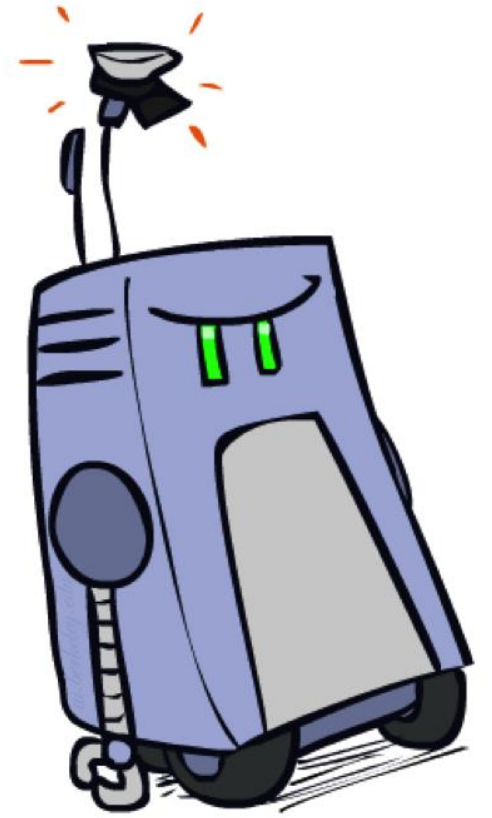
$P(\text{on}|C = 3) = 0.8$

$P(\text{on}|C = 3) = 0.9$

$P(\text{off}|C = 3) = 0.7$

$P(\text{on}|C = 3) = 0.0$

2 wins!!



Example: Overfitting

- Posterior determined by *relative* probabilities (odds ratios):

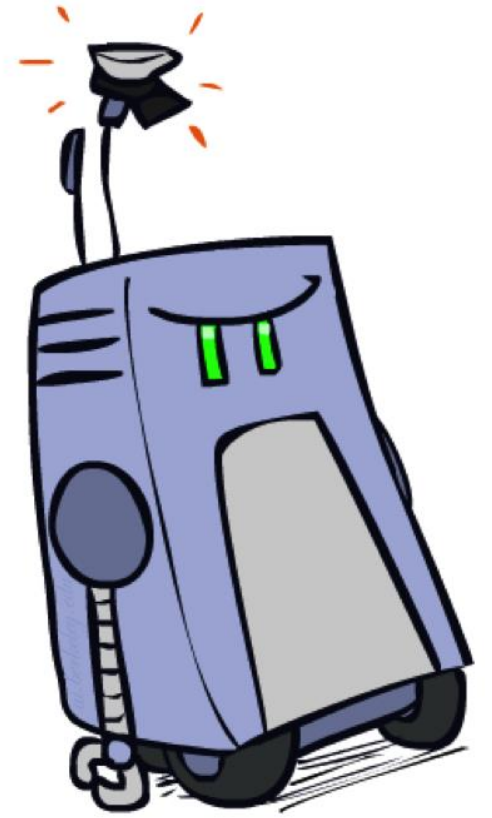
$$\frac{P(W|\text{ham})}{P(W|\text{spam})}$$

south-west	: inf
nation	: inf
morally	: inf
nicely	: inf
extent	: inf
seriously	: inf
...	

$$\frac{P(W|\text{spam})}{P(W|\text{ham})}$$

screens	: inf
minute	: inf
guaranteed	: inf
\$205.00	: inf
delivery	: inf
signature	: inf
...	

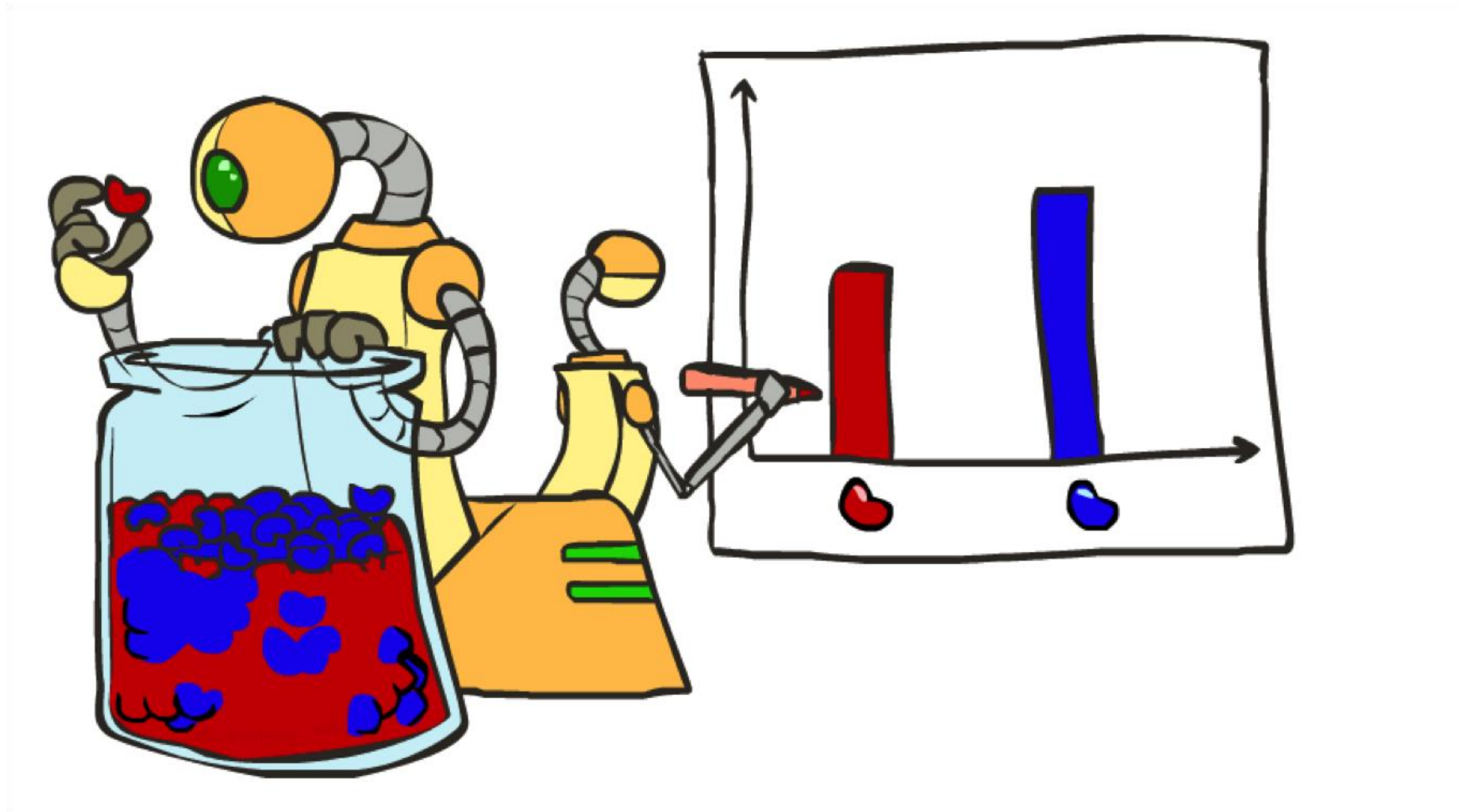
What went wrong here?



Generalization and Overfitting

- Relative frequency parameters will **overfit** the training data!
 - Just because we never saw a 3 with pixel (15,15) on during training doesn't mean we won't see it at test time
 - Unlikely that every occurrence of "minute" is 100% spam
 - Unlikely that every occurrence of "seriously" is 100% ham
 - What about all the words that don't occur in the training set at all?
 - In general, we can't go around giving unseen events zero probability
- As an extreme case, imagine using the entire email as the only feature
 - Would get the training data perfect (if deterministic labeling)
 - Wouldn't *generalize* at all
 - Just making the bag-of-words assumption gives us some generalization, but isn't enough
- To generalize better: we need to **smooth** or **regularize** the estimates

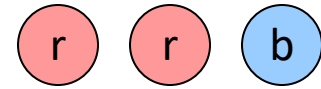
Parameter Estimation



Parameter Estimation

- Estimating the distribution of a random variable
- *Elicitation*: ask a human (why is this hard?)
- *Empirically*: use training data (learning!)
 - E.g.: for each outcome x , look at the *empirical rate* of that value:

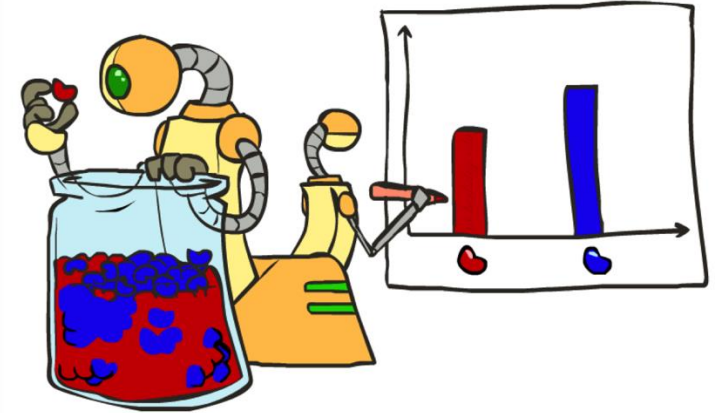
$$P_{\text{ML}}(x) = \frac{\text{count}(x)}{\text{total samples}}$$



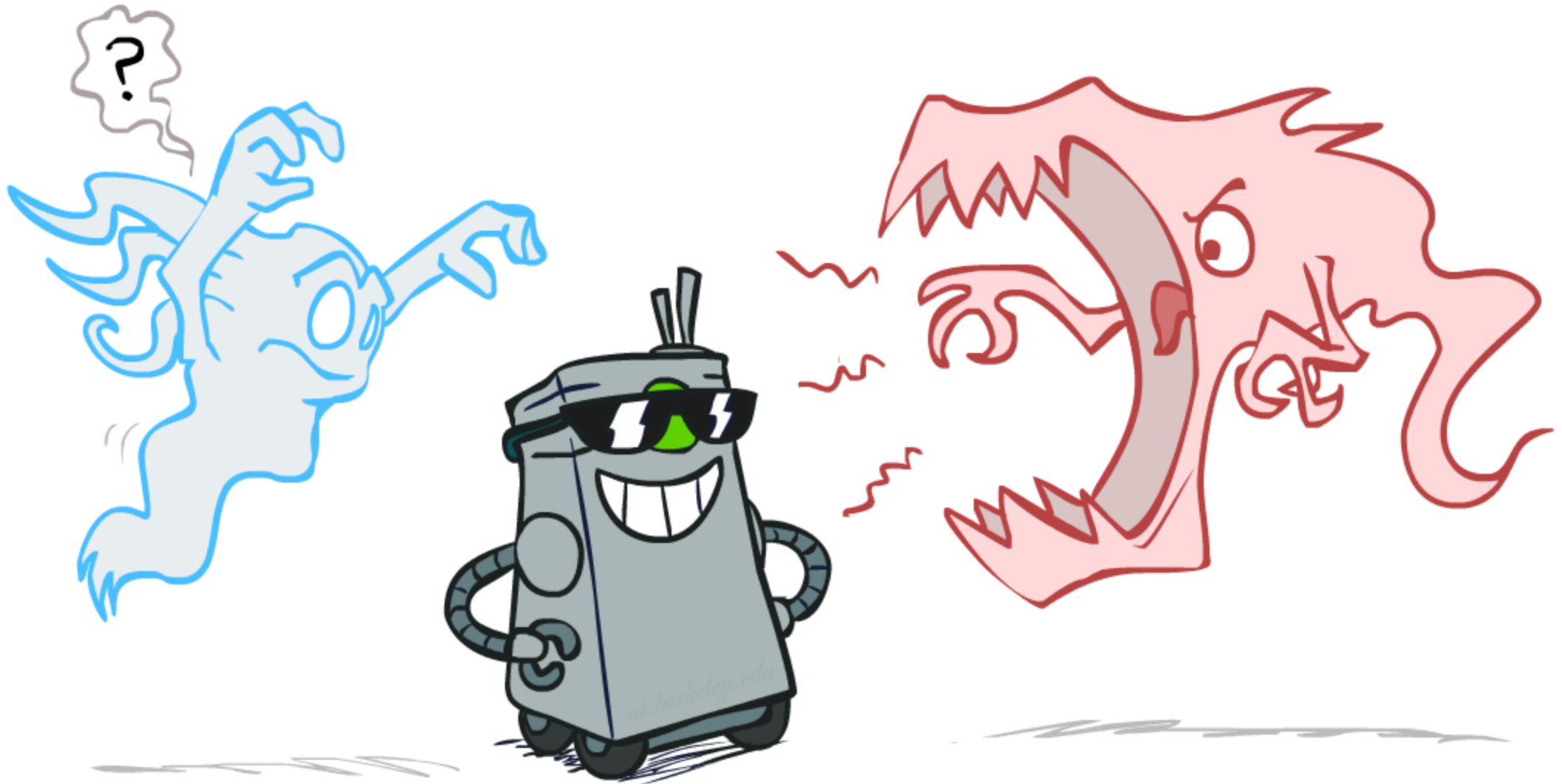
$$P_{\text{ML}}(r) = 2/3$$

- This is the estimate that maximizes the *likelihood of the data*

$$L(x, \theta) = \prod_i P_{\theta}(x_i)$$



Smoothing



Maximum Likelihood?

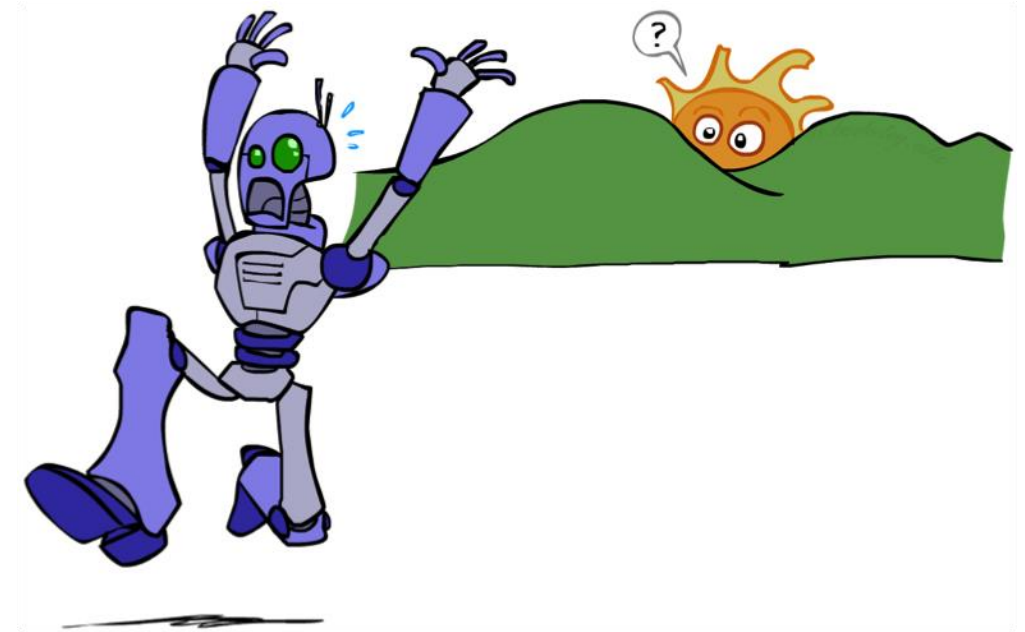
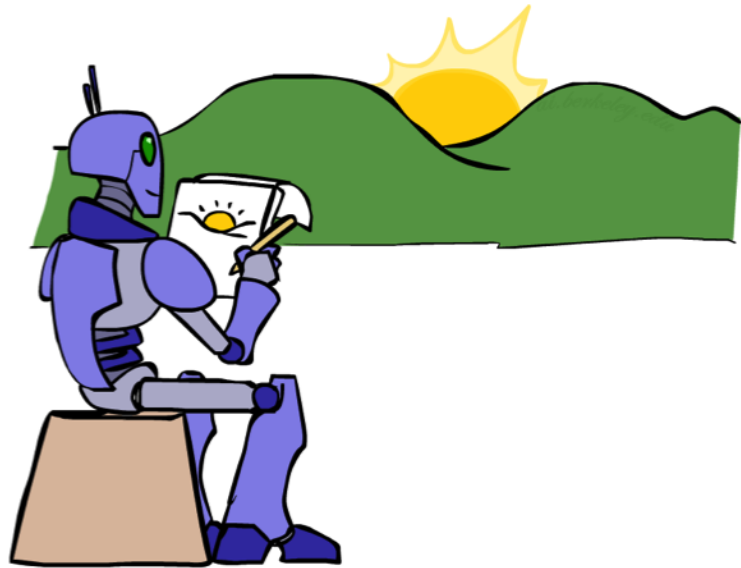
- Relative frequencies are the maximum likelihood estimates

$$\begin{aligned}\theta_{ML} &= \arg \max_{\theta} P(\mathbf{X}|\theta) \\ &= \arg \max_{\theta} \prod_i P_{\theta}(X_i)\end{aligned} \quad \Rightarrow \quad P_{ML}(x) = \frac{\text{count}(x)}{\text{total samples}}$$

- Another option is to consider the most likely parameter value given the data

$$\begin{aligned}\theta_{MAP} &= \arg \max_{\theta} P(\theta|\mathbf{X}) \\ &= \arg \max_{\theta} P(\mathbf{X}|\theta)P(\theta)/P(\mathbf{X}) \quad \Rightarrow \quad \text{????} \\ &= \arg \max_{\theta} P(\mathbf{X}|\theta)P(\theta)\end{aligned}$$

Unseen Events



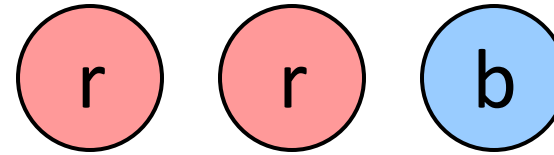
Laplace Smoothing

- Laplace's estimate:

- Pretend you saw every outcome once more than you actually did

$$\begin{aligned} P_{LAP}(x) &= \frac{c(x) + 1}{\sum_x [c(x) + 1]} \\ &= \frac{c(x) + 1}{N + |X|} \end{aligned}$$

- Can derive this estimate with *Dirichlet priors* (see cs281a)



$$P_{ML}(X) =$$

$$P_{LAP}(X) =$$

Laplace Smoothing

- Laplace's estimate (extended):

- Pretend you saw every outcome k extra times

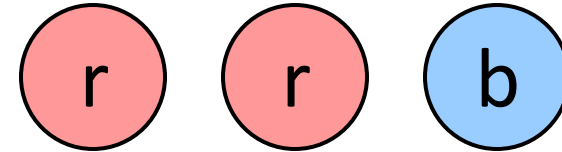
$$P_{LAP,k}(x) = \frac{c(x) + k}{N + k|X|}$$

- What's Laplace with $k = 0$?
- k is the **strength** of the prior

- Laplace for conditionals:

- Smooth each condition independently:

$$P_{LAP,k}(x|y) = \frac{c(x, y) + k}{c(y) + k|X|}$$



$$P_{LAP,0}(X) =$$

$$P_{LAP,1}(X) =$$

$$P_{LAP,100}(X) =$$

Estimation: Linear Interpolation*

- In practice, Laplace often performs poorly for $P(X|Y)$:
 - When $|X|$ is very large
 - When $|Y|$ is very large
- Another option: linear interpolation
 - Also get the empirical $P(X)$ from the data
 - Make sure the estimate of $P(X|Y)$ isn't too different from the empirical $P(X)$

$$P_{LIN}(x|y) = \alpha \hat{P}(x|y) + (1.0 - \alpha) \hat{P}(x)$$

- What if α is 0? 1?
- For even better ways to estimate parameters, as well as details of the math, see cs281a, cs288

Real NB: Smoothing

- For real classification problems, smoothing is critical
- New odds ratios:

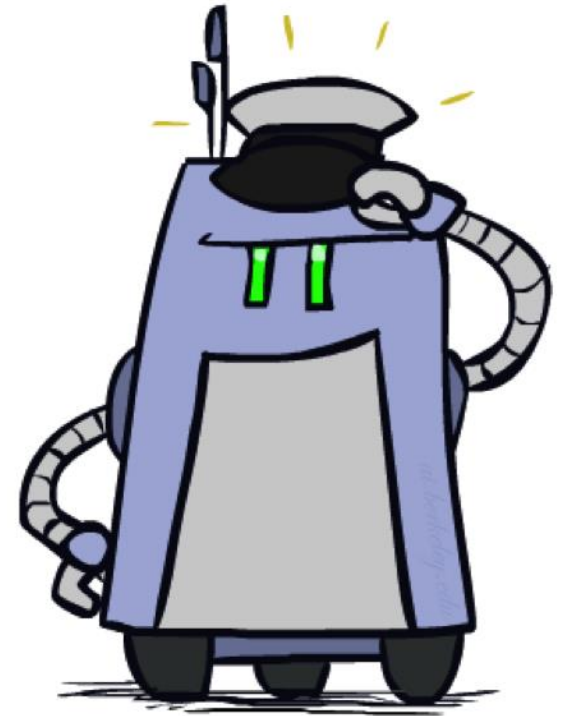
$$\frac{P(W|\text{ham})}{P(W|\text{spam})}$$

helvetica	:	11.4
seems	:	10.8
group	:	10.2
ago	:	8.4
areas	:	8.3
...		

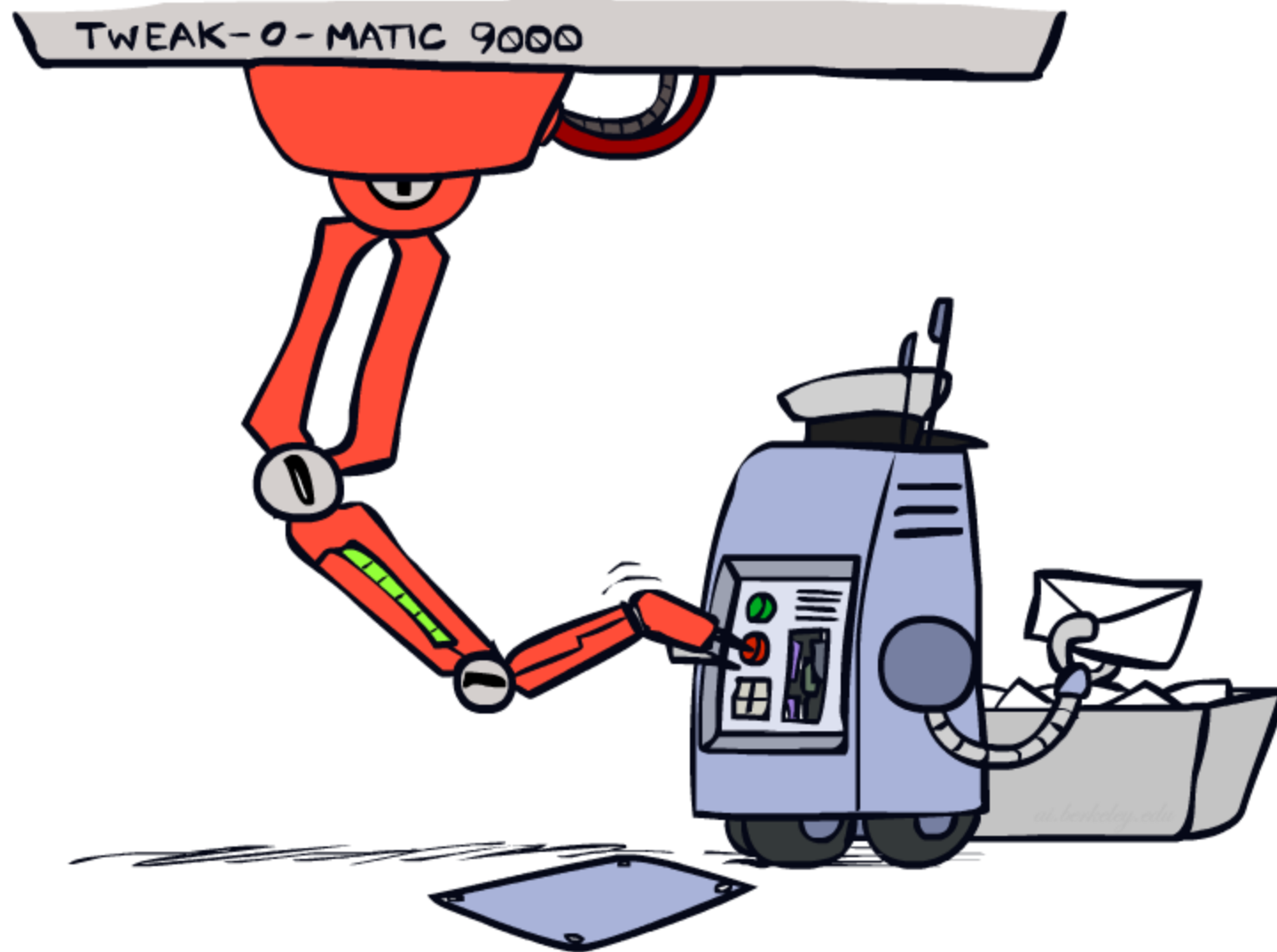
$$\frac{P(W|\text{spam})}{P(W|\text{ham})}$$

verdana	:	28.8
Credit	:	28.4
ORDER	:	27.2
	:	26.9
money	:	26.5
...		

Do these make more sense?

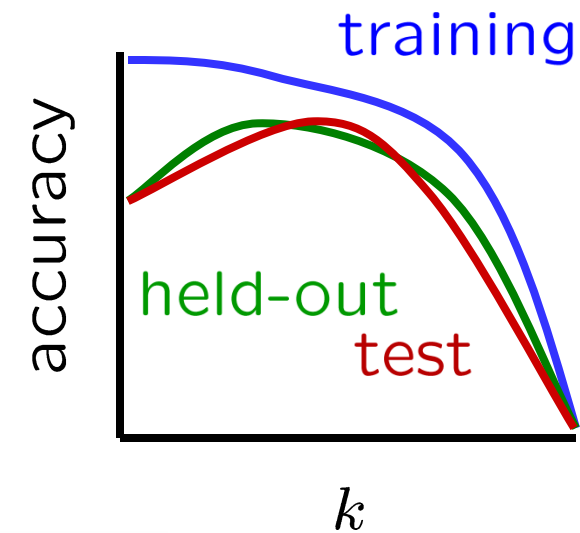


Tuning

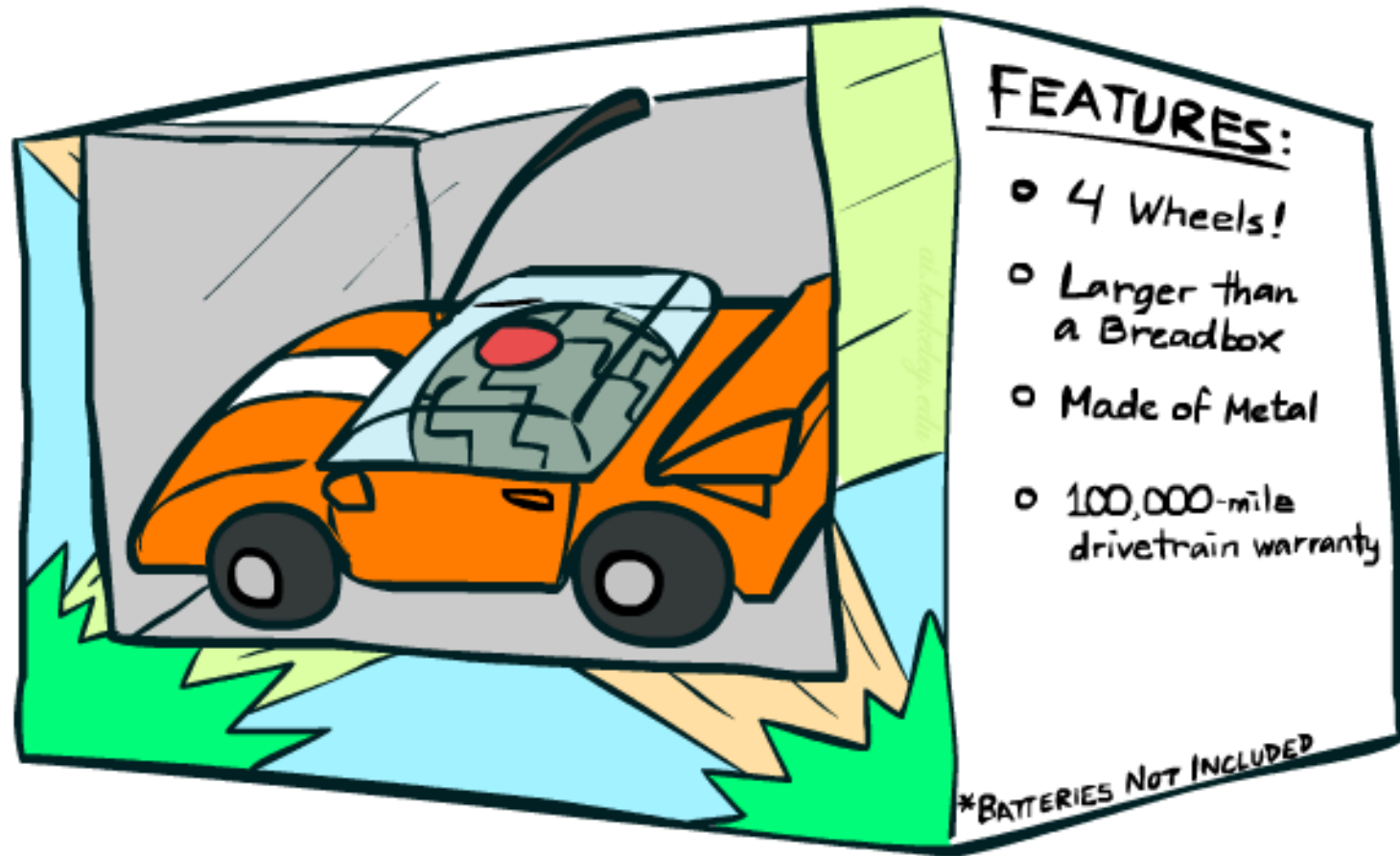


Tuning on Held-Out Data

- Now we've got two kinds of unknowns
 - Parameters: the probabilities $P(X|Y)$, $P(Y)$
 - Hyperparameters: e.g. the amount / type of smoothing to do, k , α
- What should we learn where?
 - Learn parameters from training data
 - Tune hyperparameters on different data
 - Why?
 - For each value of the hyperparameters, train and test on the held-out data
 - Choose the best value and do a final test on the test data



Features



Errors, and What to Do

- Examples of errors

Dear GlobalSCAPE Customer,

GlobalSCAPE has partnered with ScanSoft to offer you the latest version of OmniPage Pro, for just \$99.99* - the regular list price is \$499! The most common question we've received about this offer is - Is this genuine? We would like to assure you that this offer is authorized by ScanSoft, is genuine and valid. You can get the . . .

. . . To receive your \$30 Amazon.com promotional certificate, click through to

<http://www.amazon.com/apparel>

and see the prominent link for the \$30 offer. All details are there. We hope you enjoyed receiving this message. However, if you'd rather not receive future e-mails announcing new store launches, please click . . .

What to Do About Errors?

- Need more features— words aren't enough!
 - Have you emailed the sender before?
 - Have 1K other people just gotten the same email?
 - Is the sending information consistent?
 - Is the email in ALL CAPS?
 - Do inline URLs point where they say they point?
 - Does the email address you by (your) name?
- Can add these information sources as new variables in the NB model
- Next class we'll talk about classifiers which let you easily add arbitrary features more easily



Baselines

- First step: get a **baseline**
 - Baselines are very simple “straw man” procedures
 - Help determine how hard the task is
 - Help know what a “good” accuracy is
- Weak baseline: most frequent label classifier
 - Gives all test instances whatever label was most common in the training set
 - E.g. for spam filtering, might label everything as ham
 - Accuracy might be very high if the problem is skewed
 - E.g. calling everything “ham” gets 66%, so a classifier that gets 70% isn’t very good...
- For real research, usually use previous work as a (strong) baseline

Confidences from a Classifier

- The **confidence** of a probabilistic classifier:

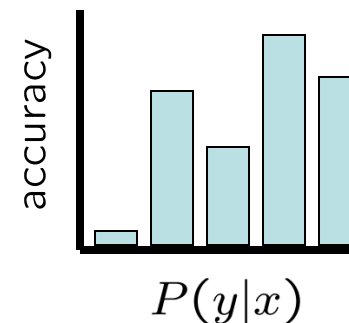
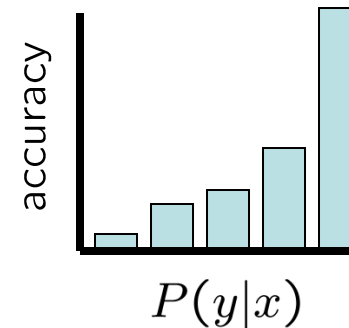
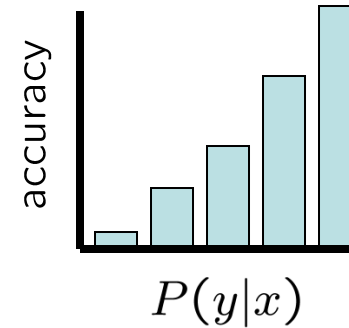
- Posterior over the top label

$$\text{confidence}(x) = \max_y P(y|x)$$

- Represents how sure the classifier is of the classification
- Any probabilistic model will have confidences
- No guarantee confidence is correct

- **Calibration**

- Weak calibration: higher confidences mean higher accuracy
- Strong calibration: confidence predicts accuracy rate
- What's the value of calibration?



Summary

- Bayes rule lets us do diagnostic queries with causal probabilities
- The naïve Bayes assumption takes all features to be independent given the class label
- We can build classifiers out of a naïve Bayes model using training data
- Smoothing estimates is important in real systems
- Classifier confidences are useful, when you can get them