

Lecture 6: Uncertainty and Probability



Gabriel Poesia

Assistant Professor, CSE

University of Michigan, Ann Arbor

poesia@umich.edu

<https://gpoesia.com>

0. Administrivia
1. Agents and uncertainty
2. Probability
3. Inference and decisions
4. Learning and independence
5. Naïve Bayes

Homework 1 check in - due this Wednesday!

Project proposal check in

Anonymous feedback form will be posted on Ed today

You will submit a Google form anonymously, then a screenshot on Gradescope of the confirmation page only
(+ 0.5% bonus point)

Will help us adjust and improve in the next modules

AGENTS AND UNCERTAINTY

Logical uncertainty

A logical agent keeps a knowledge base and asks what follows from it.

Sometimes both a proposition and its negation are possible given what we know:

$$KB \not\models P \quad \text{and} \quad KB \not\models \neg P$$

Some models of the KB might make P true, others might make it false.

So logic already lets us represent uncertainty.

But not **quantify** it... If we had to make a decision, we could try to ask:

“Which possible worlds seem most plausible?”

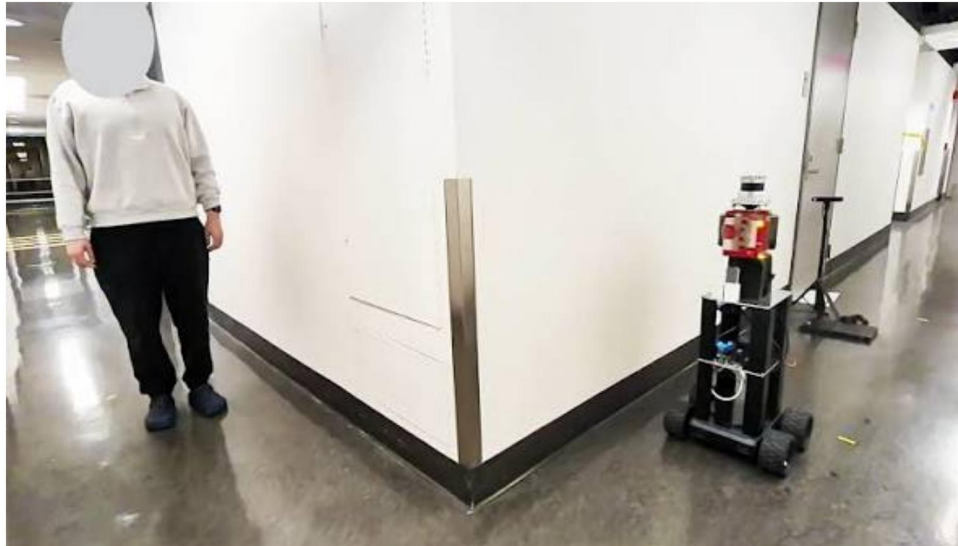
**We want to say: “I don't know whether P is true,
but the evidence makes it much more likely than $\neg P$.”**

Sources of uncertainty

So far we assumed we had the whole state and deterministic transitions.
In practice, agents can be uncertain for several reasons:

Partial observation

Example: a robot can't see around a corner



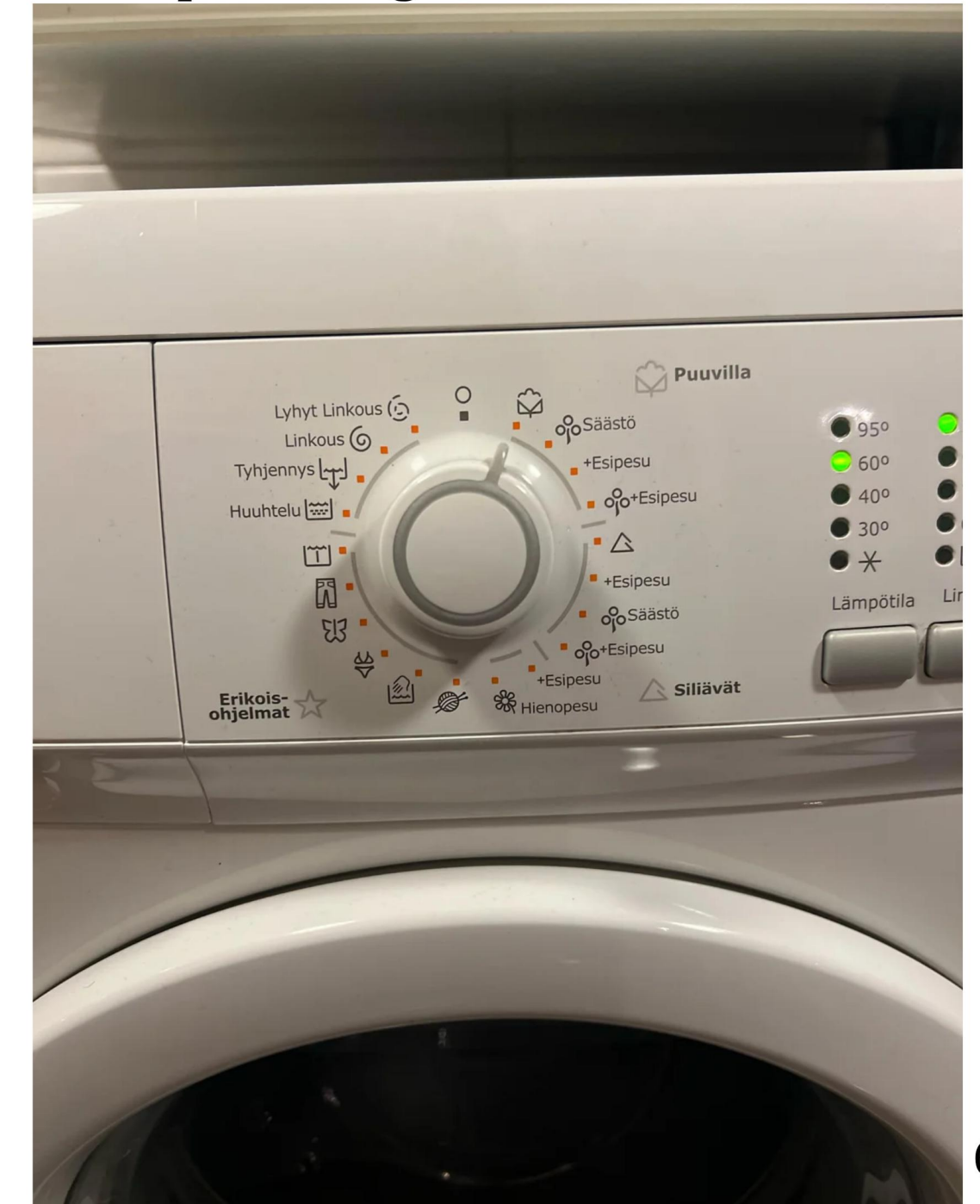
Noisy action outcomes

Example: physical actuators (and sensors) have noise



Incomplete world model

Example: using this machine

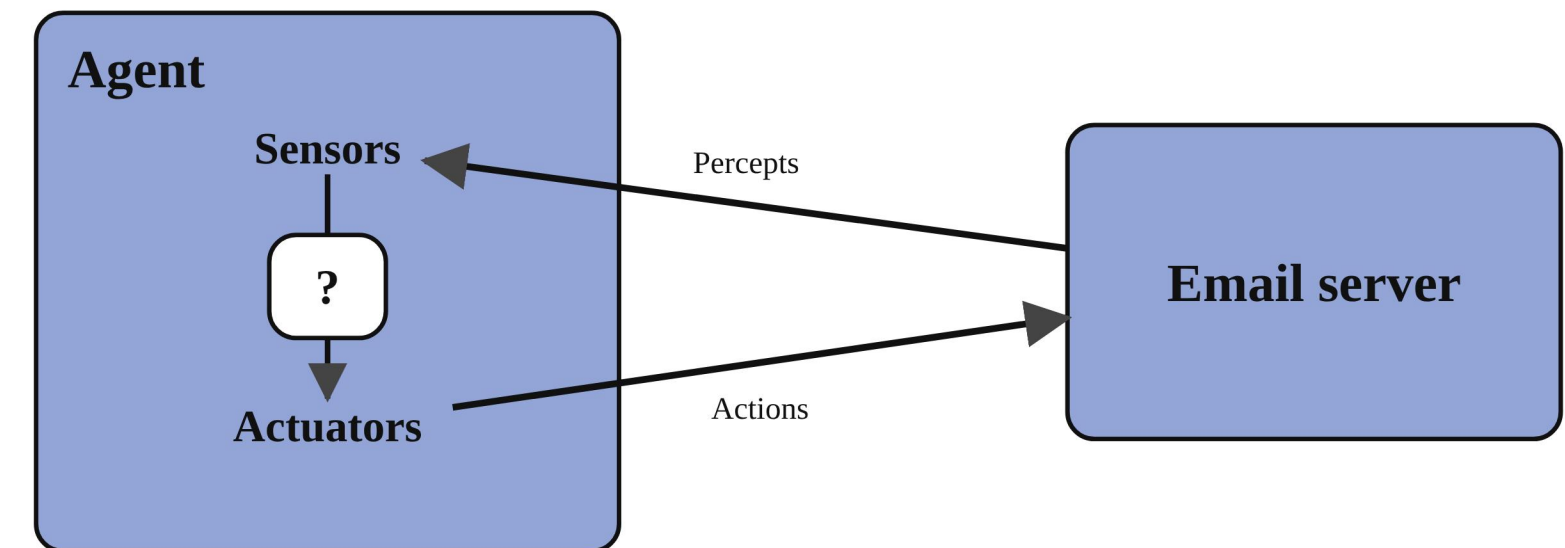


Suppose an email handling agent that chooses whether to deliver or quarantine each new email.

- **Performance:** keep spam out without hiding useful email
- **Environment:** email server and incoming messages
- **Actuators:** request next message; deliver / quarantine
- **Sensors:** message features returned by the server

We'll model this task as **episodic**: assume one choice doesn't change the future.

That is an assumption about the task. It does not mean the messages themselves are statistically independent.



PROBABILITY

Random experiments

Our agent has a button: **Get next mail.**

What happens when it presses the button?

DEFINITION (RANDOM EXPERIMENT)

A repeatable action or procedure whose outcome is uncertain.

In our simplified world, suppose for now that an email has only a few attributes:

Feature	Possible values
TLD	.edu, .com, or .xyz
Authenticated?	yes or no
Occurrences of “free”	0, 1, 2, ...
Spam?	yes or no (true in the world, but hidden from the agent)

Outcomes and the sample space

The result of pressing the button will be an **outcome**.

DEFINITION (OUTCOME)

An outcome ω is one possible result of a random experiment.

Here, an outcome includes the message's hidden class:

$$\omega = (\text{TLD, authenticated?, "free" count, spam?})$$

DEFINITION (SAMPLE SPACE)

The sample space Ω is the set of all possible outcomes.

$$\Omega = \{\text{edu, com, xyz}\} \times \{0, 1\} \times \{0, 1, 2, \dots\} \times \{0, 1\}$$

We use 1 for yes and 0 for no. The word count has no fixed upper bound in this model.

Example outcomes

Some possible outcomes of pressing the button:

	TLD	Authenticated?	“free” count	Spam? (hidden)
ω_1	.edu	yes	0	no
ω_2	.com	no	1	yes
ω_3	.xyz	no	4	yes
ω_4	.com	yes	0	no
ω_5	.edu	no	2	no

Containing "free" makes it *more likely* spam, but it's not guaranteed!

We often reason about sets of many possible outcomes when we say sentences like “The next message is spam” (regardless of TLD, etc).

We call such a set an **event**.

DEFINITION (EVENT)

An event is a set of outcomes: $A \subseteq \Omega$.

Some examples:

Email

Some possible events:

S : the hidden Spam field is "yes"

E : the TLD is .edu

$S \cap E$: spam from a .edu address

$\neg S$: next message is not spam (Spam = "no")

Rolling two dice

The outcome is the ordered pair of numbers:

$$\Omega = \{1, \dots, 6\}^2.$$

Event “Sum is 7” is the set:

$$\{(1, 6), (2, 5), (3, 4), (4, 3), (5, 2), (6, 1)\}$$

$$\text{Event “Doubles”}: \{(1, 1), (2, 2), \dots, (6, 6)\}$$

Probability axioms

How likely is a given event to happen? This is given by its **probability**.

DEFINITION (PROBABILITY MEASURE)

A probability measure assigns a number $P(A)$ to each event, satisfying three axioms:

1. $P(A) \geq 0$
2. $P(\Omega) = 1$
3. For disjoint events $A_1, A_2, \dots$:
 $P(\bigcup_i A_i) = \sum_i P(A_i)$

For example, $P(S) = 0.20$ says that the next message has a 20% chance of being spam.

If we roll two fair dice independently, each outcome (an ordered pair) has probability $1/36$.

There are six mutually exclusive ways for their sum to be 7, so:

$$P(\text{sum is 7}) = 6 \cdot \frac{1}{36} = \frac{1}{6}$$

Probability identities

These rules imply several identities we'll use repeatedly:

$$P(\emptyset) = 0, \quad P(\neg A) = 1 - P(A)$$

So $P(S) = 0.20$ implies $P(\neg S) = 0.80$.

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

Let F be the event that the message mentions “free.” Suppose:

$$P(S) = 0.20, \quad P(F) = 0.16, \quad P(S \cap F) = 0.12$$

$$P(S \cup F) = 0.20 + 0.16 - 0.12 = 0.24$$

We subtract the overlap because it was counted twice. Also, $A \subseteq B$ implies $P(A) \leq P(B)$.

Random variables

Sometimes we only receive (or choose to consider) a function of the true, raw outcome.
Example: perhaps we might care only about the *sum* of the dice instead of both numbers.
Such a function of outcomes is called a random variable.

DEFINITION (RANDOM VARIABLE)

A random variable X is a function from outcomes to a set of values: $X : \Omega \rightarrow \mathcal{X}$.

Different outcomes can produce the same value. For example:

$$X(1, 6) = X(2, 5) = \dots = X(6, 1) = 7$$

$\mathcal{X}$ is the variable's range. For the sum, $\mathcal{X} = \{2, \dots, 12\}$.

Despite the name, a random variable is a deterministic function. The randomness comes from which outcome occurs.

Probability distributions

Experiment	Sample space Ω	Random variable	Range $\mathcal{X}$
Roll two dice	Ordered pairs of faces	X : sum	$\{2, \dots, 12\}$
Roll two dice	Ordered pairs of faces	M : maximum	$\{1, \dots, 6\}$
Sample a student	Students in the class	H : height in inches	$\mathbb{R}_{>0}$
Get next mail	Email records	F : mentions “free”?	$\{0, 1\}$

Statements such as $X = 7$ or $65 \leq H \leq 70$ define events. A **distribution** gives their probabilities.

Discrete: probability mass function (PMF)

$p_X(x) = P(X = x) \geq 0$, with $\sum_x p_X(x) = 1$.
We add masses $p_X(x)$ to find probabilities such as $P(X \leq 5)$.

Continuous: probability density function (PDF)

For continuous range (e.g. height):
 $P(a \leq H \leq b) = \int_a^b f_H(h) dh$.
Probability is the area under the density.
Has to satisfy $f_H(h) \geq 0$ and $\int f_H(h) dh = 1$.

INFERENCE AND DECISIONS

For now, our bot will consider only two Boolean random variables:

- S : is the message spam?
- F : does the message contain “free”?

Both take values in $\{0, 1\}$. We'll write S for the event $S = 1$, and $\neg S$ for $S = 0$; likewise for F . Each variable has its own PMF. To describe the two variables *together*, we use a joint distribution.

DEFINITION (JOINT DISTRIBUTION)

A joint distribution assigns a probability to every complete combination of values:

$$P(X_1 = x_1, \dots, X_n = x_n).$$

	F	$\neg F$
S	0.12	0.08
$\neg S$	0.04	0.76

Each cell is the probability of *both* its row and its column. For example, $P(S, F) = P(S \cap F) = 0.12$. The four cells describe mutually exclusive, exhaustive possibilities. They sum to 1.

Suppose we know the joint. How do we recover the distribution of just one variable?

DEFINITION (MARGINALIZATION)

Marginalizing out Y means obtaining a distribution over the remaining variables by summing the joint distribution over every possible value of Y :

$$P(X = x) = \sum_y P(X = x, Y = y).$$

Why? If Y can be $y_0, y_1, \dots$, then $X = x$ is the union of $(X = x, Y = y_0), (X = x, Y = y_1), \dots$ (all disjoint since Y is different).

For our messages, sum a row to get $P(S)$, or a column to get $P(F)$:

	F	$\neg F$	Total
S	0.12	0.08	0.20
$\neg S$	0.04	0.76	0.80
Total	0.16	0.84	1

The column contains two disjoint events, so $P(F) = P(S, F) + P(\neg S, F) = 0.16$.

Conditional probability

Suppose we learn that an event B occurred. How likely is another event A now?

DEFINITION (CONDITIONAL PROBABILITY)

For $P(B) > 0$:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}.$$

Roll two fair dice independently. Let A be “the sum is at least 6” and B be “the first die is 1.”

B contains six outcomes: $(1, 1), (1, 2), \dots, (1, 6)$.

Only $(1, 5)$ and $(1, 6)$ also belong to A .

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \frac{2/36}{6/36} = \frac{1}{3}$$

Conditioning **restricts** us to the outcomes in B , then **renormalizes** their probabilities to sum to 1.

Conditional distributions

Let's go back to the joint distribution of spam label S and the observed word feature F :

	F	$\neg F$	Total
S	0.12	0.08	0.20
$\neg S$	0.04	0.76	0.80
Total	0.16	0.84	1

Suppose we observe F (email has word "free"). We can now just focus on the first column, whose total mass is $P(F) = 0.16$, and renormalize:

Spam

$$P(S \mid F) = 0.12/0.16 = 0.75$$

Not spam

$$P(\neg S \mid F) = 0.04/0.16 = 0.25$$

Together these probabilities form the conditional distribution $P(S \mid F)$, and they sum to 1.

Is $P(F \mid S)$ also 0.75?

$$P(F \mid S) = \frac{P(F, S)}{P(S)} = \frac{0.12}{0.20} = 0.60$$

Among messages containing “free,” 75% are spam. Among spam messages, 60% contain “free.” Different questions!

Law of total probability

A message containing “free” can be spam or not spam. Marginalization gave us:

$$P(F) = P(F, S) + P(F, \neg S)$$

From the conditional probability we get $P(F, S) = P(F | S)P(S)$.

$$P(F) = P(F | S)P(S) + P(F | \neg S)P(\neg S)$$

So the overall rate is a weighted average of the rates within the two classes.

DEFINITION (LAW OF TOTAL PROBABILITY)

If $B_1, \dots, B_k$ partition the sample space and $P(B_i) > 0$, then

$$P(A) = \sum_{i=1}^k P(A | B_i)P(B_i).$$

A partition covers every outcome exactly once. Here it is $\{S, \neg S\}$.

Bayes' rule

For brevity, we write $P(A, B)$ to mean $P(A \cap B)$.

Recall the definition of conditional probability:

$$P(A \mid B) = \frac{P(A, B)}{P(B)}$$

Rearranging gives the **product rule**:

$$P(A, B) = P(A \mid B)P(B)$$

But the same joint probability also equals $P(B \mid A)P(A)$.

THEOREM (BAYES' RULE)

Assuming $P(B) > 0$, we have

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}.$$

$P(A)$ is the **prior**: our probability for A before learning B .

$P(B \mid A)$ says how likely the new evidence B is if A holds. Dividing by $P(B)$ normalizes the result.

$P(A \mid B)$ is the **posterior**: our probability for A after learning B .

Suppose our first bot just flags all messages containing “free.” We know:

- $P(S) = 0.20$
- $P(F | S) = 0.60$
- $P(F | \neg S) = 0.05$

Suppose some message was flagged. What is the probability it's spam?

$$\begin{aligned} P(S | F) &= \frac{P(F|S)P(S)}{P(F)} = \frac{P(F|S)P(S)}{P(F|S)P(S) + P(F|\neg S)P(\neg S)} \\ &= \frac{0.60 \cdot 0.20}{0.60 \cdot 0.20 + 0.05 \cdot 0.80} = \frac{0.12}{0.16} = 0.75 \end{aligned}$$

The denominator includes **both** ways F can happen in terms of S (spam and not spam).

What if only 1% of messages were spam, with the same $P(F|S)$ and $P(F|\neg S)$?

$$\text{Now, } P(S | F) = \frac{0.60 \cdot 0.01}{0.60 \cdot 0.01 + 0.05 \cdot 0.99} \approx 0.108$$

The **base rate** matters a lot (here, $P(S)$)! This can be counter-intuitive.

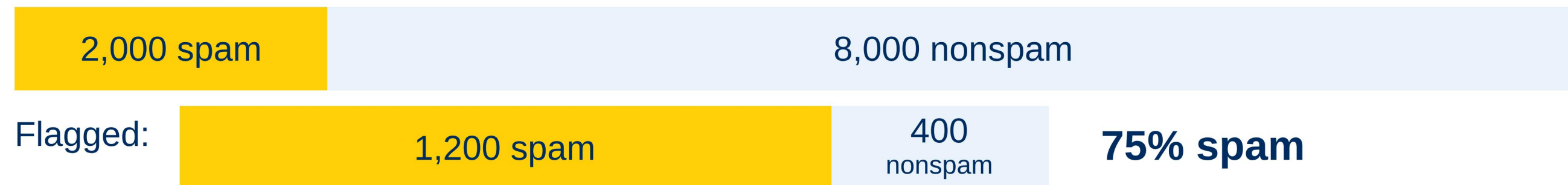
Base rates: a counting example

The flagging rates are the same in both populations:

$$P(F \mid S) = 0.60, \quad P(F \mid \neg S) = 0.05$$

20% base rate

10,000 messages



1% base rate

10,000 messages



Seeing "free" is much less convincing when spam is very rare!

Classical example here is with tests for extremely rare diseases (low "prevalence", say 1 millionth of population).

Even if test is 99.9% accurate, if you test positive, it might be more likely you are a false positive case.

Probabilistic inference

Usually, an agent observes only some variables and asks about others.

Query Q

What we want to know: spam?

Evidence $E = e$

What we observed: “free” present

Hidden variables H

What we didn't observe: sender type, topic, etc.

If we have the joint distribution, we can answer by summing and normalizing:

$$P(Q = q \mid E = e) = \frac{\sum_h P(q, e, h)}{\sum_{q', h} P(q', e, h)}$$

The denominator is $P(e)$; assume it is positive.

This is **inference**: using a given model to answer a query.

We will now look at: **learning** the model from data.

The full joint table is easy to do inference on, but we'll see it will be hard to learn.

Expected utility

Last note before we do learning: even with inference, our agent still has to choose an action!

Say inference gives our message is spam with some probability p . Should we send it to quarantine?

That depends on how much we mind each kind of mistake.

This is typically formalized as an utility function U that assigns values to outcomes of our actions.

We can then choose actions to maximize our *expected utility*.

Example: say a correct decision has utility 0, letting spam pass has utility -1 , and quarantining good email is 10 times worse.

Suppose inference tells us: "spam with probability $p = 0.75$ ". What is our expected utility for each action?

Action	Spam	Not spam	Expected utility
Deliver	-1	0	-0.75
Quarantine	0	-10	$-10 \cdot 0.25 = -2.5$

Which action should we choose in this model?

Average each action's utility over each possible result, weighted by their probabilities. Delivering is better: $-0.75 > -2.5$.

If spam with probability 95%, then -0.95 vs -0.5 : quarantine is better in expectation!

You can imagine the user (or system designer) sets the utility, user preferences could change this

This is the essence of *decision making under uncertainty*: how to go from probabilities to actions

We'll come back to this topic later in the course

LEARNING AND INDEPENDENCE

Coin identification

So far we were given the probabilities. Can we *learn* them from many observations?

Let's start with a smaller problem: suppose a friend gives us one of two coins without saying which:

Coin A

Fair: $P(H) = 0.5$

Coin B

Biased: $P(H) = 0.75$

Say we've done this before, and saw they pick A 80% of the time and B 20%. These are our **prior probabilities**.

Supppose we flip the coin five times in a row, and get:

$$D = HHHHT$$

Which coin do we think was chosen?

A was more likely to be picked, but four heads might favor B. How do we combine those two pieces of information?

Write θ for the coin's probability of heads. Our two hypotheses are $\theta = 0.5$ and $\theta = 0.75$.

For the observed sequence $D = HHHHT$, we can calculate how likely is that outcome given each θ :

$$P(D \mid \theta) = \theta \cdot \theta \cdot \theta \cdot \theta \cdot (1 - \theta) = \theta^4(1 - \theta)$$

This is a function of the parameter θ known as the likelihood.

DEFINITION (LIKELIHOOD)

The likelihood $L(\theta) = P(D \mid \theta)$ is the probability of the observed data, viewed as a function of the parameter θ .

Coin A

$$L(0.5) = (1/2)^5 = 32/1024$$

Coin B

$$L(0.75) = (3/4)^4(1/4) = 81/1024$$

The sequence is about 2.5 times as likely under B.

That is evidence for B, but it is not yet the probability that we picked B.

Remember $P(A) = 0.8$, $P(B) = 0.2$, and we observed $D = HHHHT$.

Bayes' rule now answers a question about the unknown coin:

$$P(\theta \mid D) = \frac{P(D|\theta)P(\theta)}{\sum_{\theta'} P(D|\theta')P(\theta')}$$

Coin	Prior	Likelihood	Prior $\times$ likelihood	Posterior
A: $\theta = 0.5$	4/5	32/1024	128/5120	128/209 $\approx$ 0.612
B: $\theta = 0.75$	1/5	81/1024	81/5120	81/209 $\approx$ 0.388

The normalizer is $P(D) = (128 + 81)/5120$. Divide each row's weight by this total.

B's probability rose from 20% to about 39%. A is still more likely.

Bayesian learning updates a distribution over hypotheses.

If we later have more observations, this posterior becomes the prior for our next updates.

Our posterior gives probabilities to both coins. Suppose we want to give our best guess, $\hat{\theta}$. One intuitive strategy is choosing the estimate with highest posterior

DEFINITION (MAXIMUM A POSTERIORI (MAP) ESTIMATE)

Among our discrete candidates, choose a parameter value with the highest posterior probability:

$$\hat{\theta}_{\text{MAP}} \in \arg \max_{\theta} P(\theta \mid D) = \arg \max_{\theta} P(D \mid \theta)P(\theta).$$

Note we can ignore denominator $P(D)$ we'd get from Bayes' rule. Why?

It's the same for every candidate, so it doesn't change the max!

Here $128/209 > 81/209$, so $\hat{\theta}_{\text{MAP}} = 0.5$: guess Coin A.

Assuming the prior is correct, this choice minimizes our probability of making a mistake.

Maximum likelihood estimation

What if we don't want to assume a prior?

We could also just have picked based on the likelihood alone.

DEFINITION (MAXIMUM LIKELIHOOD ESTIMATION (MLE))

Among the allowed parameter values Θ , choose one that gives the observed data the highest probability:

$$\hat{\theta}_{\text{ML}} \in \arg \max_{\theta \in \Theta} P_{\theta}(D).$$

P_{θ} means the distribution with parameter θ . We evaluate every candidate on the same observed data D .

For our two coins, $81/1024 > 32/1024$, so MLE chooses B.

Estimator	What do we maximize?	Guess for $HHHT$
MAP	Likelihood $\times$ prior	A: $\theta = 0.5$
MLE	Likelihood	B: $\theta = 0.75$

With a uniform prior over these candidates, the two methods would agree.

Example: coin with arbitrary bias

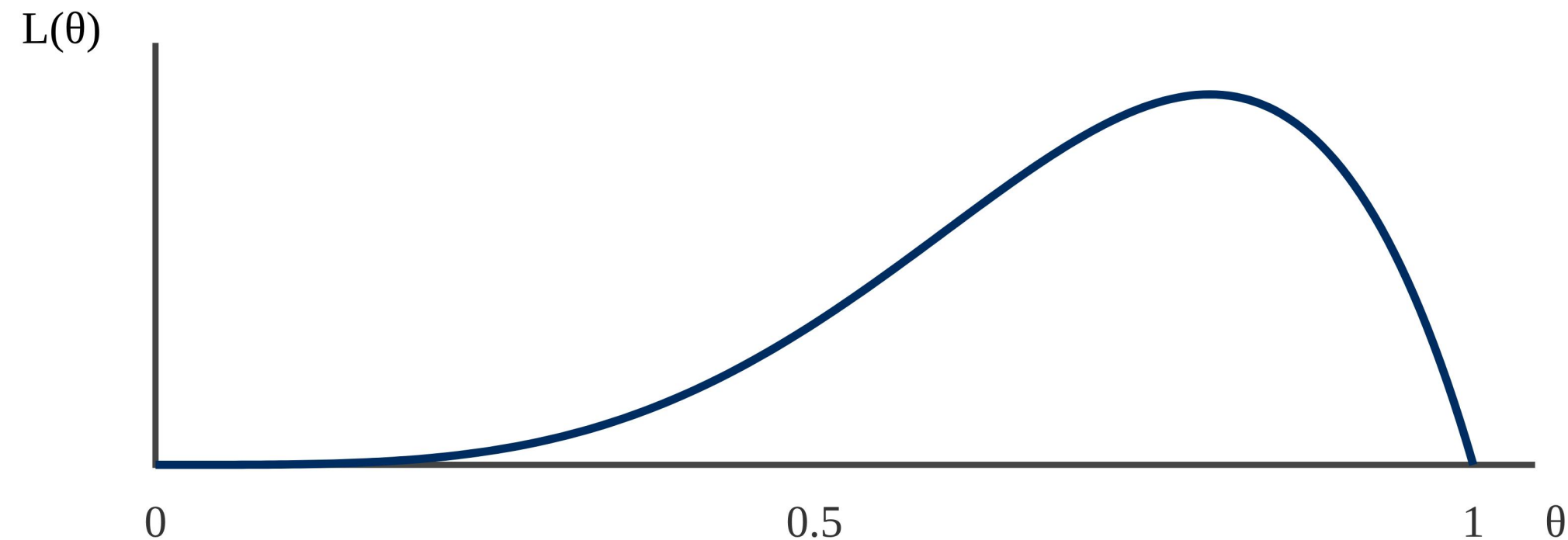
What if this could have been any coin with any bias?

Suppose it could be any $\theta \in [0, 1]$.

Suppose the same observations: $HHHHT$. The likelihood is still:

$$L(\theta) = \theta^4(1 - \theta)$$

As a function of θ , this looks like:



Small θ makes the four heads unlikely. What about $\theta = 1$?

$\theta = 1$ makes the T impossible! So likelihood becomes 0

What θ maximizes the likelihood? Let's work it out.

For observations drawn independently from the same distribution P_θ , the likelihood is a product:

$$L(\theta) = \prod_{i=1}^n P_\theta(x_i)$$

We often want to find the maximum by differentiating and setting to zero.

Products are hard to differentiate, and multiplying many small probabilities can underflow with floats.

The logarithm turns the product into a sum:

DEFINITION (LOG-LIKELIHOOD)

Add the log probabilities of the observations under the candidate parameter:

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^n \log P_\theta(x_i).$$

log is strictly increasing: larger positive likelihood means larger log-likelihood. So the optimum is the same.

$$HHHHT : \quad \ell(\theta) = 4 \log \theta + \log(1 - \theta)$$

Optimizing the coin likelihood

For $0 < \theta < 1$, differentiate the log-likelihood and set its derivative to zero:

$$\ell'(\theta) = \frac{4}{\theta} - \frac{1}{1-\theta} = 0$$

$$4(1 - \theta) = \theta \implies 4 = 5\theta \implies \hat{\theta}_{\text{ML}} = \frac{4}{5}$$

Is this a maximum? We can test with the second derivative:

$$\ell''(\theta) = -\frac{4}{\theta^2} - \frac{1}{(1-\theta)^2} < 0$$

So the slope keeps decreasing, and this stationary point is a maximum.

This is just the observed (empirical) fraction of heads!

In fact, in general with h heads in $n > 0$ flips, the same argument gives $\hat{\theta} = h/n$.

(Seeing all heads or all tails would put the maximum at the extremes, either 1 or 0)

Categorical MLE

Let's come back to the email server.

Suppose we want to use MLE to estimate the fraction of emails that come from each TLD:

Let's say the true (unobserved) rates are $\theta_{\text{edu}}, \theta_{\text{com}}, \theta_{\text{xyz}}$

We don't know these, but we see emails. Suppose we see ten messages with these TLDs:

TLD	.edu	.com	.xyz
Count	6	3	1

There are now three parameters, constrained to sum to one:

$$\ell(\theta) = 6 \log \theta_{\text{edu}} + 3 \log \theta_{\text{com}} + \log \theta_{\text{xyz}}$$

Maximizing this gives the empirical rates, just as for the coin.

So MLE would give $\hat{\theta}_{\text{edu}} = 0.6, \hat{\theta}_{\text{com}} = 0.3, \hat{\theta}_{\text{xyz}} = 0.1$

DEFINITION (CATEGORICAL MLE)

For a variable with k possible values, let n_j be the count of value j in $n > 0$ independent samples from the same distribution. Then, MLE gives

$$\hat{P}(X = j) = \frac{n_j}{n}.$$

We can estimate the true rates from data by counting.

Note that if we just happen to not see a certain value, we get probability zero (we'll return to this).

Joint distribution size

Could we learn the whole joint distribution this way?

In principle, could treat each combination of values as a different category and count frequencies.

In general, if variable X_i has r_i possible values, the joint table has:

$$\prod_{i=1}^d r_i \text{ entries}$$

If we had 30 Boolean features, we'd have 2^{30} entries (more than a billion)

Each of these is one *parameter* we're learning (minus 1 because of normalization)

This is bad for at least two reasons:

1. Large to store and query.
2. We need data to estimate these entries, too: see each combination possibly many times to estimate frequencies.

Can we use structure in the distribution to help?

The structure that will help is *independence*

Example: for two dice rolled independently, seeing the first face shouldn't change the distribution of the second.

DEFINITION (INDEPENDENCE)

X and Y are independent, written $X \perp Y$, if

$$P(X = x, Y = y) = P(X = x)P(Y = y) \quad \text{for every } x, y.$$

Even for biased dice, we can learn two six-entry PMFs and multiply them to recover all 36 joint entries.

For many variables, **mutual independence** means the entire joint factors:

$$P(x_1, \dots, x_d) = \prod_{i=1}^d P(x_i)$$

We only need to learn each marginal distribution. The number of free parameters is:

$$\sum_{i=1}^d (r_i - 1)$$

For d Boolean variables, that is d parameters instead of $2^d - 1$.

Often independence doesn't actually hold, but is a convenient *modeling assumption*

How well it will work depends on how reasonable the assumption is in the actual scenario.

Conditional independence

Let's go back to the hidden coin. If we don't know the bias, are the outcomes independent?

No! After seeing 4 heads and no tails, another head becomes more likely (we have evidence for a biased coin).

But if we already know the coin's bias θ , the first flip tells us nothing more about the second.

DEFINITION (CONDITIONAL INDEPENDENCE)

X and Y are conditionally independent given Z , written $X \perp Y \mid Z$, if for all z with $P(z) > 0$,

$$P(x, y \mid z) = P(x \mid z)P(y \mid z) \quad \text{for all } x, y.$$

Here Z is the chosen coin, and X, Y are two flip outcomes.

Let's see one *actually useful* method for classifying spam by heavily assuming independence.

Naïve Bayes

Bag of words

Real messages of course contain more than a few small features.

For instance, there is the text itself!

For now, suppose we assume a dictionary with $V = 100,000$ words.

A simple representation for a message is the number of times each dictionary word occurs:

$$c = (c_1, \dots, c_V), \quad c_w = \text{count of word } w$$

For “free free winner,” $c_{\text{free}} = 2$, $c_{\text{winner}} = 1$, and every other count is zero.

This is the **bag of words** model: “winner free free” has the same representation.

Let S be the hidden class: spam or non-spam. Given a labeled dataset, we want to learn:

$$P(S = s \mid c)$$

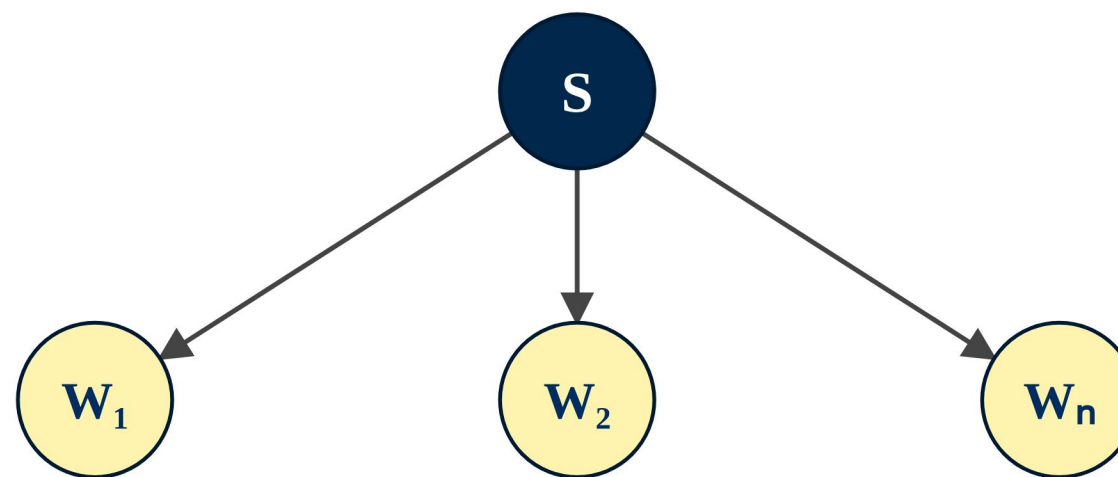
Instead of counting how often we saw this exact message, we'll learn which words each class tends to use.

Naïve Bayes for text

For a message of length n , imagine the following model of how messages are generated:

1. First choose a class s according to $P(S)$.
2. Then, draw each of the n words independently from the same class-specific distribution.

Let a **token** be just one occurrence of a word. W_i is the word at position i ; $\theta_{w,s} = P(W_i = w \mid S = s)$.



The shared class can make words dependent overall.

Example: if we $W_1 = \text{"free"}$, maybe $W_2 = \text{"money"}$ becomes more likely, $W_2 = \text{"expensive"}$ less likely

But given the class, we're assuming words are independent.

This is an example of completely "wrong" model (no email is actually generated like this) but still useful!

Multinomial Naïve Bayes

Under our model, the probability of the sequence “free free winner” given class s and $n = 3$ is:

$$\theta_{\text{free},s} \cdot \theta_{\text{free},s} \cdot \theta_{\text{winner},s} = \theta_{\text{free},s}^2 \theta_{\text{winner},s}$$

For a bag of words, add the probabilities of every ordering with those counts.

DEFINITION (MULTINOMIAL NAÏVE BAYES)

Assuming $n = \sum_w c_w$, the conditional distribution of word counts is

$$P(c \mid s) = \frac{n!}{\prod_w c_w!} \prod_{w=1}^V \theta_{ws}^{c_w}, \quad \theta_{ws} \geq 0, \quad \sum_w \theta_{ws} = 1.$$

For this example, there are $3!/2! = 3$ orderings. But that factor is the same for spam and non-spam!

Bayes' rule therefore gives a simple score to compare classes:

$$P(s \mid c) \propto \text{score}(s|c) = P(s) \prod_{w:c_w>0} \theta_{ws}^{c_w}$$

Same observation as in MAP: we don't need to know the normalizing constant

Naïve Bayes training

For learning, we usually assume a *training* dataset where we know the label (e.g., we had humans label a set of emails)
Suppose we have N training messages, each labeled spam or non-spam. What do we count?

- N_s : number of training messages with class s
- C_{ws} : total occurrences of word w in messages with class s
- $T_s = \sum_w C_{ws}$: total word tokens in messages with class s

$$\hat{P}(s) = \frac{N_s}{N}, \quad \hat{\theta}_{ws} = \frac{C_{ws}}{T_s}$$

The prior counts *messages*. The word distributions count *tokens*: “free free” is two occurrences.
For example, suppose part of our count table looks like this:

Class	Messages N_s	Tokens T_s	“free”	“meeting”	“winner”
Spam	40	1000	100	5	50
non-spam	60	2000	10	100	2

(Plus other 99,997 word columns not shown)

Naïve Bayes prediction

Now say we get a two-word message: “**free winner**”.

From the training counts:

Class	Prior	$P(\text{free} \mid s)$	$P(\text{winner} \mid s)$	Score
Spam	0.4	100/1000	50/1000	0.002
non-spam	0.6	10/2000	2/2000	0.000003

The spam score is larger, so the model predicts spam.

To obtain probabilities, normalize the scores:

$$\hat{P}(S \mid c) = \frac{0.002}{0.002 + 0.000003} \approx 0.9985$$

The numerator alone is a score, not a posterior. Dividing by the sum makes the class probabilities add to one.

Naïve Bayes complexity

How many parameters does our model have now?

For two classes, Naïve Bayes only stores two distributions over the dictionary: so $2V$ word probabilities, plus the class prior.

Training is just one pass over the labeled dataset counting classes and word tokens.

For the score, like with log-likelihood, we can avoid multiplying tiny numbers by adding their logarithms:

$$\log \text{score}(s|c) = \log P(s) + \sum_{w:c_w>0} c_w \log \theta_{ws}$$

Most of the 100,000 counts are zero. We only visit words that actually occur in the message.

For k classes and n words in the message, scoring takes $O(kn)$. Parameters take $O(kV)$ space.

One shared distribution per class is crucial: we don't learn a different dictionary distribution at every word position.

Laplace smoothing

Suppose we had never seen the word "GPU" in spam, and we suddenly get: "free GPU now".
The count of "GPU" in spam during training will be 0, so:

$$C_{\text{GPU},S} = 0 \implies \hat{\theta}_{\text{GPU},S} = 0$$

Then any future message containing that word gets spam score zero, regardless of all its other words.
Not seeing a word in a finite dataset should not make it impossible.

DEFINITION (LAPLACE SMOOTHING)

We can smooth our count data and avoid predicting probability 0 by adding an $\alpha > 0$ to the count of every word in every class:

$$\tilde{\theta}_{ws} = \frac{C_{ws} + \alpha}{T_s + V\alpha}.$$

With $\alpha = 1$, $T_s = 1000$ and $V = 100,000$, an unseen word receives probability $1/101,000$.
The denominator adds $V\alpha$ because we added one α count to each of the V possible words.
Using these smoothed probabilities when scoring avoids predicting exactly 0

Modeling assumptions

Naïve Bayes is surprisingly accurate in many practical cases, and extremely efficient to train and run.

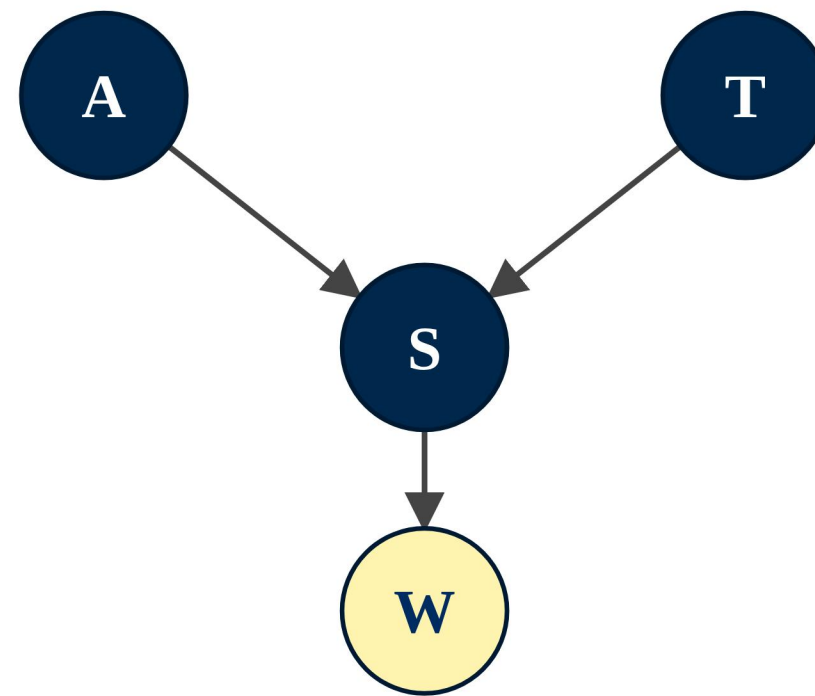
Our model relied on several assumptions:

- **Representative data:** training email should resemble email we expect in the future
- **Bag of words:** we ignored word order and didn't use email length as evidence
- **Conditional independence:** given the class, each token was drawn independently from the same word distribution

In many cases, we don't want to throw away *all dependencies*, some might be very important

Bayesian networks

Suppose spam status S depends on authentication A and TLD T , while a word feature W depends on S .



This particular graph is assuming $A \perp T$, but then S depends on both A and T .

$$P(A, T, S, W) = P(A)P(T)P(S \mid A, T)P(W \mid S)$$

A **Bayesian network** uses a directed acyclic graph and one conditional distribution per variable:

$$P(x_1, \dots, x_n) = \prod_i P(x_i \mid \text{Parents}(X_i))$$

$\text{Parents}(X_i)$ means the variables with arrows into X_i . Here S has parents A, T , so it needs $P(S \mid A, T)$.

Next lecture: answer queries using the graph, keeping just the dependencies we want to model.

- **Probability:** assigns weights to events in a sample space
- **Inference:** we can marginalize hidden variables and condition on observed evidence
- **Actions:** combine probabilities with utilities to choose what to do
- **Learning:** update beliefs with data; estimate unknown parameters by MAP or MLE
- **Independence:** allows us to learn only marginal distributions and multiply them to recover the joint
- **Naïve Bayes:** text classifier using just class and word counts
- **Smoothing:** we generally don't want to say something we haven't seen is completely impossible!

We can now build simple agents that learn from examples and reason with uncertainty.

Next: Bayesian networks, where we use a graph to represent our dependency structure.