# AI Programming

## *CS662-2013S-16*

## *More Probability Theory*

David Galles

Department of Computer Science
University of San Francisco

# 16-0: Probability Review

- Probability allows us to represent a belief about a statement, or a likelihood that a statement is true.
  - $P(rain) = 0.6$ means that we believe it is 60% likely that it is currently raining.
- Axioms:
  - $0 \leq P(a) \leq 1$
  - The probability of $(A \vee B)$ is $P(A) + P(B) - P(A \wedge B)$
  - Tautologies have $P = 1$
  - Contradictions have $P = 0$

# 16-1: Conditional Probability

- Once we begin to make observations about the value of certain variables, our belief in other variables changes.
  - Once we notice that it's cloudy, $P(rain)$ goes up.
- this is called *conditional probability*
- Written as: $P(rain|cloudy)$
- $P(a|b) = \frac{P(a \wedge b)}{P(b)}$
- or $P(a \wedge b) = P(a|b)P(b)$
  - This is called the *product rule*.

# 16-2: Conditional Probability

- Example: $P(Cloudy) = 0.25$
- $P(Rain) = 0.25$
- $P(cloudy \wedge rain) = 0.15$
- $P(cloudy \wedge \neg Rain) = 0.1$
- $P(\neg cloudy \wedge Rain) = 0.1$
- $P(\neg Cloudy \wedge \neg Rain) = 0.65$
  - Initially, $P(Rain) = 0.25$. Once we see that it's cloudy, $P(Rain|Cloudy) = P\frac{(Rain \wedge Cloudy)}{P(Cloudy)} = \frac{0.15}{0.25} = 0.6$

# 16-3: Combinations of events

- The probability of $(A \wedge B)$ is $P(A|B)P(B)$
- What if $A$ and $B$ are independent?
- Then $P(A|B)$ is $P(A)$, and $P(A \wedge B)$ is $P(A)P(B)$.
- Example:
  - What is the probability of “heads” five times in a row?
  - What is the probability of at least one “head”?

# 16-4: Bayes' Rule

- Often, we want to know how a probability changes as a result of an observation.
- Recall the Product Rule:
  - $P(a \wedge b) = P(a|b)P(b)$
  - $P(a \wedge b) = P(b|a)P(a)$
- We can set these equal to each other
  - $P(a|b)P(b) = P(b|a)P(a)$
- And then divide by $P(a)$
  - $P(b|a) = \frac{P(a|b)P(b)}{P(a)}$
- This equality is known as Bayes' theorem (or rule or law).

# 16-5: Bayes' Rule

- We can generalize Bayes' rule, by adding in some more evidence:
  - $P(b|a) = \frac{P(a|b)P(b)}{P(a)}$
  - $P(b|a,e) = \frac{P(a|b,e)P(b|e)}{P(a|e)}$

# 16-6: Bayes' Rule

- We can also avoid the pesky $P(a)$:
  - $P(b|a) = \frac{P(a|b)P(b)}{P(a)}$
  - $P(b|a) = \alpha P(a|b)P(b)$
- Where $\alpha$ is a normalizing constant, so that $P(b|a)$ and $P(\neg b|a)$ sum to 1.
  - Generally, so that $\sum_{b \in B} P(b|a)$ sums to 1
- Example:
  - $P(t|d) = 0.8, P(\neg t|\neg d) = 0.9, P(d) = 0.2$
  - $P(d|t)$ ?

# 16-7: Bayes' Rule

- $P(t|d) = 9/10, P(\neg t|\neg d) = 8/10, P(d) = 1/10$

$$
\begin{aligned}
P(d|t) &= \alpha P(t|d)P(d) \\
&= \alpha 9/10 * 1/10 \\
&= \alpha 9/100 \\
P(\neg d|t) &= \alpha P(t|\neg d)P(\neg d) \\
&= \alpha 2/10 * 9/10 \\
&= \alpha 18/100 \\
\alpha &= \frac{1}{9/100 + 18/100} = \frac{100}{27}
\end{aligned}
$$

# 16-8: Bayes' Rule

- $P(t|d) = 9/10, P(\neg t|\neg d) = 8/10, P(d) = 1/10$

$$
\begin{aligned}
P(d|t) &= \alpha P(t|d)P(d) \\
&= \alpha 9/10 * 1/10 \\
&= \alpha(9/100) \\
&= (100/27)(9/100) \\
&= 1/3 \\
P(\neg d|t) &= \alpha P(t|\neg d)P(\neg d) \\
&= \alpha 2/10 * 9/10 \\
&= \alpha(18/100) \\
&= (100/27)(18/100) \\
&= 2/3
\end{aligned}
$$

# 16-9: More Probability

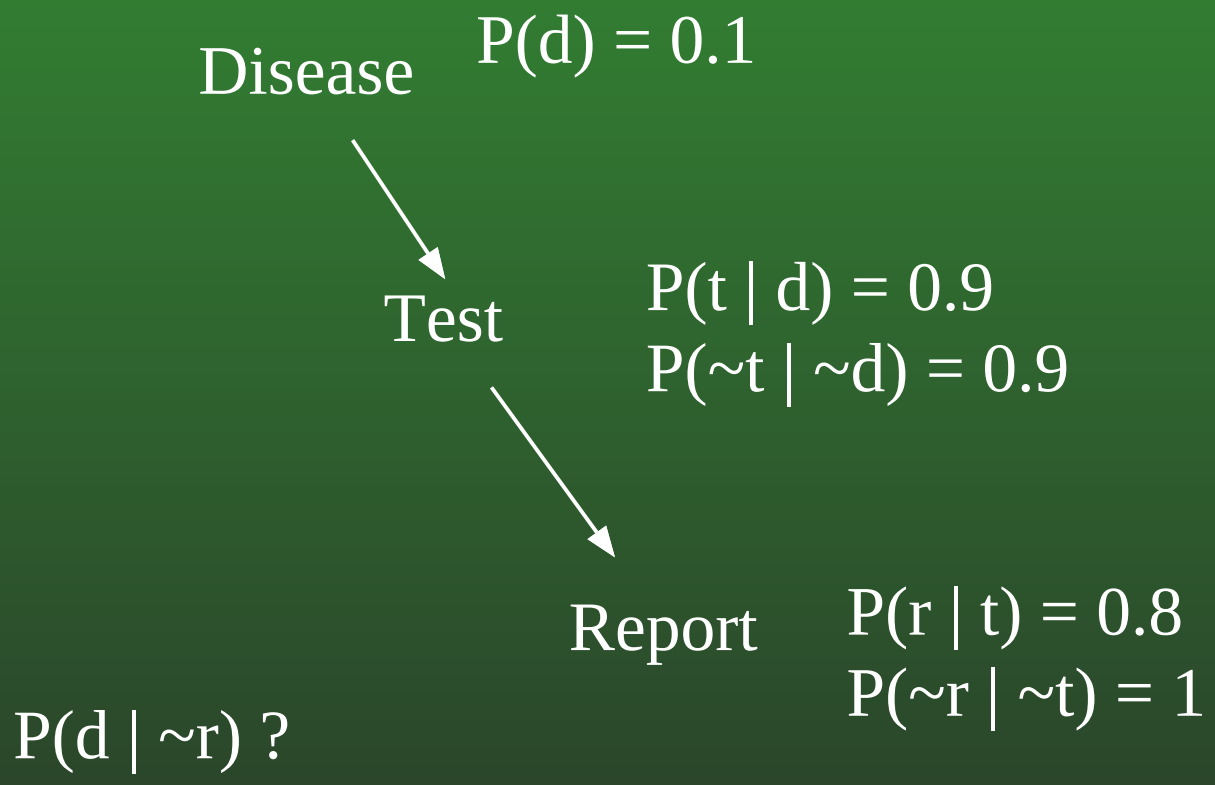
$$P(a) = \sum_{b \in B} P(a|b)P(b)$$

- Intuitively: It's always either day or night (and never both). Probability of of something happening is the probability that it happens at night, plus the probability that it happens during the day

$$P(a|e) = \sum_{b \in B} P(a|b, e)P(b|e)$$

- For any probability function, we can always condition *everything* for some additional evidence

# 16-10: Example



# 16-11: Example

- $P(d|\neg r) = \alpha P(\neg r|d)P(d)$
  - We know $P(d)$, we just need $P(\neg r|d)$ and $P(\neg r)$.
  - $P(\neg r|d)$ We only know $P(R)$ in terms of $T$ ...
- $P(\neg r|d) = P(\neg r|d,t)P(t|d) + P(\neg r|d,\neg t)P(\neg t|d)$
  - Conditional Independence to the rescue!
  - $P(\neg r|d,t) = P(\neg r|t)$, which we know.

# 16-12: Example

$$
\begin{aligned}
P(d|\neg r) &= \alpha(P(\neg r|t)P(t|d) + P(\neg r|\neg t)P(\neg t|d)) * P(d) \\
&= \alpha((0.2) * (0.9) + (1) * 0.1) * 0.1 \\
&= \alpha * 0.019
\end{aligned}
$$

$$
\begin{aligned}
P(\neg d|\neg r) &= \alpha(P(\neg r|t)P(t|\neg d) + P(\neg r|\neg t)P(\neg t|\neg d)) * P(\neg d) \\
&= \alpha((0.2) * (0.1) + (1) * 0.9) * 0.9 \\
&= \alpha(0.828)
\end{aligned}
$$

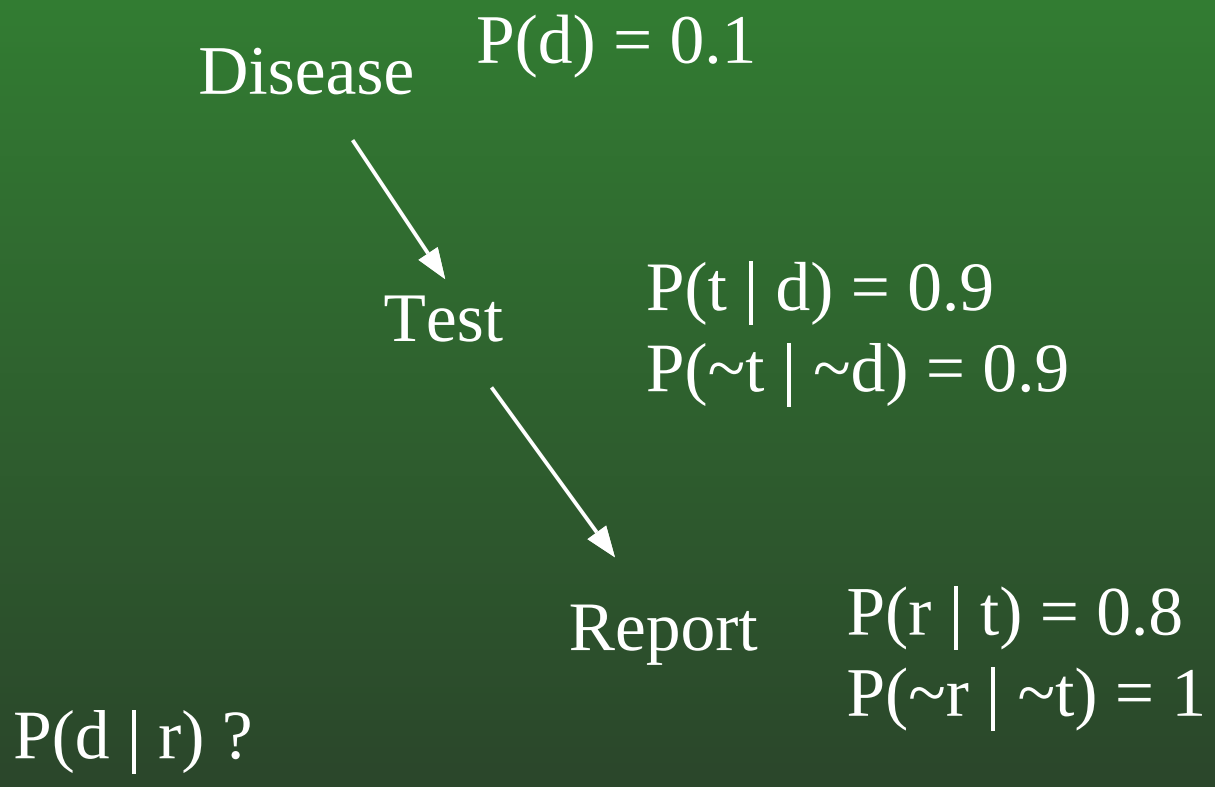
$$
\alpha = \frac{1}{0.828 + 0.019} = \frac{1}{0.847}
$$

# 16-13: Example

$$
\begin{aligned}
P(d|\neg r) &= \alpha(P(\neg r|t)P(t|d) + P(\neg r|\neg t)P(\neg t|d)) * P(d) \\
&= \alpha((0.2) * (0.9) + (1) * 0.1) * 0.1 \\
&= \alpha * 0.019 \\
&= 0.022
\end{aligned}
$$

$$
\begin{aligned}
P(\neg d|\neg r) &= \alpha(P(\neg r|t)P(t|\neg d) + P(\neg r|\neg t)P(\neg t|\neg d)) * P(\neg d) \\
&= \alpha((0.2) * (0.1) + (1) * 0.9) * 0.9 \\
&= \alpha(0.828) \\
&= 0.978
\end{aligned}
$$

# 16-14: Example II



# 16-15: Example

- $P(d|r) = \alpha P(r|d)P(d)$
  - $P(r|d) = P(r|d,t)P(t|d) + P(r|d,\neg t)P(\neg t|d)$
    - Conditional Independence to the rescue!
    - $P(r|d,t) = P(r|t)$, which we know.

# 16-16: Example

$$
\begin{aligned}
P(d|r) &= \alpha(P(r|t)P(t|d) + P(r|\neg t)P(\neg t|d)) * P(d) \\
&= \alpha((0.8) * (0.9) + (0) * 0.1) * 0.1 \\
&= \alpha * 0.072
\end{aligned}
$$

$$
\begin{aligned}
P(\neg d|r) &= \alpha(P(r|t)P(t|\neg d) + P(r|\neg t)P(\neg t|\neg d)) * P(\neg d) \\
&= \alpha((0.8) * (0.1) + (0) * 0.9) * 0.9 \\
&= \alpha(0.072)
\end{aligned}
$$

$P(d|r) = P(\neg d|r) = 0.5$

# 16-17: Learning and Classification

- An important sort of learning problem is the *classification* problem.
- This involves placing examples into one of two or more classes.
  - Should/shouldn't play tennis
  - Spam/not spam.
  - Wait/don't wait at a restaurant
- Classification is a *supervised* learning task.
  - Requires access to a set of labeled training examples
- From this we induce a hypothesis that describes how to determine what class an example should be in.

# 16-18: Probabilistic Learning

- Decision trees are one way to do this.
  - They tell us the most likely classification for given data.
  - Work best with tabular data, where each attribute has a known number of possible values.
- What if we want to know how likely a hypothesis is?
- We can apply our knowledge of probability to learn a hypothesis.

# 16-19: Bayes' Theorem

- Recall the definition of Bayes' Theorem
- $P(b|a) = \frac{P(a|b)P(b)}{P(a)}$
- Let's rewrite this a bit.
- Let $D$ be the data we've seen so far.
- Let $h$ be a possible hypothesis
- $P(h|D) = \frac{P(D|h)P(h)}{P(D)}$

# 16-20: MAP Hypothesis

- Often, we're not so interested in the particular probabilities for each hypothesis.
- Instead, we want to know: Which hypothesis is most likely, given the data?
  - Which classification is the most probable?
  - Is $PlayTennis$ or $\neg PlayTennis$ more likely?
- We call this the *maximum a posteriori hypothesis* (MAP hypothesis).
- In this case, we can ignore the denominator $(P(D))$ in Bayes' Theorem, since it will be the same for all $h$.
- $h_{MAP} = argmax_{h \in H} P(D|h)P(h)$

# 16-21: MAP Hypothesis

- Advantages:
  - Simpler calculation
  - No need to have a prior for $P(D)$

# 16-22: ML Hypothesis

- In some cases, we can simplify things even further.
- What are the priors $P(h)$ for each hypothesis?
- Without any other information, we'll often assume that they're equally possible.
  - Each has probability $\frac{1}{H}$
- In this case, we can just consider the conditional probability $P(D|h)$.
- We call the hypothesis that maximizes this conditional probability the *maximum likelihood* hypothesis.
- $h_{ML} = argmax_{h \in H} P(D|h)$

# 16-23: Example

- Imagine that we have a large bag of candy. We want to know the ratio of cherry to lime in the bag.
- We start with 5 hypotheses:
  1. $h_1$: 100% cherry
  2. $h_2$ 75% cherry, 25% lime.
  3. $h_3$ 50% cherry, 50% lime
  4. $h_4$ 25% cherry, 75% lime
  5. $h_5$ 100% lime
- Our agent repeatedly draws pieces of candy.
- We want it to correctly pick the type of the next piece of candy.

# 16-24: Example

- Let's assume our priors for the different hypotheses are:
- $(0.1, 0.2, 0.4, 0.2, 0.1)$
- Also, we assume that the observations are i.i.d.
  - Independent and Identically Distributed – each choice is independent of the others, and order doesn't matter.
- In that case, we can multiply probabilities.
- $P(D|h_i) = \Pi_j P(d_j|h_i)$
- Suppose we draw 10 limes in a row. $P(D|h_3)$ is $(\frac{1}{2})^{10}$, since the probability of drawing a lime under $h_3$ is $\frac{1}{2}$.

# 16-25: Example

- How do the hypotheses change as data is observed?
- Initially, we start with the priors: $(0.1, 0.2, 0.4, 0.2, 0.1)$
- Then we draw a lime.
  - $P(h_1|lime) = \alpha P(lime|h_1)P(h_1) = 0.$
  - $P(h_2|lime) = \alpha P(lime|h_2)P(h_2) = \alpha\frac{1}{4} * 0.2 = \alpha 0.05.$
  - $P(h_3|lime) = \alpha P(lime|h_3)P(h_3) = \alpha\frac{1}{2} * 0.4 = \alpha 0.2$
  - $P(h_4|lime) = \alpha P(lime|h_4)P(h_4) = \alpha\frac{3}{4} * 0.2 = \alpha 0.15.$
  - $P(h_5|lime) = \alpha P(lime|h_5)P(h_5) = \alpha 1 * 0.1 = \alpha 0.1.$
  - $\alpha = 2.$

# 16-26: Example

- Then we draw a second lime.
  - $P(h_1|lime, lime) = \alpha P(lime, lime|h_1)P(h_1) = 0.$
  - $P(h_2|lime, lime) = \alpha P(lime, lime|h_2)P(h_2) = \alpha \frac{1}{4}\frac{1}{4} * 0.2 = \alpha 0.0125.$
  - $P(h_3|lime, lime) = \alpha P(lime, lime|h_3)P(h_3) = \alpha \frac{1}{2}\frac{1}{2} * 0.4 = \alpha 0.1$
  - $P(h_4|lime, lime) = \alpha P(lime, lime|h_4)P(h_4) = \alpha \frac{3}{4}\frac{3}{4} * 0.2 = \alpha 0.1125.$
  - $P(h_5|lime) = \alpha P(lime|h_5)P(h_5) = \alpha 1 * 0.1 = \alpha 0.1.$
  - $\alpha = 3.07.$
- Strictly speaking, we don't really care what $\alpha$ is.
- We can just select the MAP hypothesis, since we just want to know the most likely hypothesis.

# 16-27: Bayesian Learning

- Eventually, the true hypothesis will dominate all others.
  - Caveat: assuming the data is noise-free, or noise is uniformly distributed.
- Notice that we can use Bayesian learning (in this case) either as a batch algorithm or as an incremental algorithm.
- We can always easily update our hypotheses to incorporate new evidence.
  - This depends on the assumption that our observations are independent.

# 16-28: Learning bias

- What sort of bias does Bayesian Learning use?
- Typically, simpler hypotheses will have larger priors.
- More complex hypotheses will fit data more exactly (but there's many more of them).
  - Under these assumptions, $h_{MAP}$ will be the simplest hypothesis that fits the data.
  - This is Occam's razor, again.

# 16-29: Bayesian Concept Learning

- Bayesian Learning involves estimating the likelihood of each hypothesis.
- In a more complex world where observations are not independent, this could be difficult.
- Our first cut at doing this might be a brute force approach:
  1. For each $h$ in $H$, calculate $P(h|D) = \frac{P(D|h)P(h)}{P(D)}$
  2. From this, output the hypothesis $h_{MAP}$ with the highest posterior probability.
- This is what we did in the example.
  - Challenge - Bayes' Theorem can be computationally expensive to use when observations are not i.i.d.

# 16-30: Bayesian Optimal Classifiers

- There's one other problem with the formulation as we have it.
- Usually, we're not so interested in the hypothesis that fits the data.
- Instead, we want to classify some unseen data, given the data we've seen so far.
- One approach would be to just return the MAP hypothesis.
- We can do better, though.

# 16-31: Bayesian Optimal Classifiers

- Suppose we have three hypotheses and posteriors: $h_1 = 0.4, h_2 = 0.3, h_3 = 0.3$.
- We get a new piece of data - $h_1$ says it's positive, $h_2$ and $h_3$ negative.
- $h_1$ is the MAP hypothesis, yet there's a 0.6 chance that the data is negative.
- By combining weighted hypotheses, we improve our performance.

# 16-32: Bayesian Optimal Classifiers

- By combining the predictions of each hypothesis, we get a Bayesian optimal classifier.
- More formally, let's say our unseen data belongs to one of $v$ classes.
- The probability $P(v_j|D)$ that our new instance belongs to class $v_j$ is:
- $\sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$
- Intuitively, each hypothesis gives its prediction, weighted by the likelihood that that hypothesis is the correct one.
- This classification method is provably optimal - on average, no other algorithm can perform better.

# 16-33: Problems

- However, the Bayes optimal classifier is mostly interesting as a theoretical benchmark.
- In practice, computing the posterior probabilities is exponentially hard.
- This problem arises when instances or data are conditionally dependent upon each other.
- Can we get around this?

# 16-34: Naive Bayes classifier

- The Naive Bayes classifer makes a strong assumption that makes the algorithm practical:
  - Each attribute of an example is independent of the others.
  - $P(a \wedge b) = P(a)P(b)$ for all a and b.
- This makes it straightforward to compute posteriors.

# 16-35: Bayesian Learning Problem

- Given: a set of labeled, multivalued examples.
- Find a function $F(x)$ that correctly classifies an unseen example with attributes $(a_1, a_2, ..., a_n)$ .
- Call the most probable category $v_{map}$.
- $v_{map} = argmax_{v_i \in V} P(v_i | a_1, a_2, ..., a_n)$
- We rewrite this with Bayes' Theorem as:
  $v_{map} = argmax_{v_i \in V} P(a_1, a_2, ..., a_n | v_i) P(v_i)$
- Estimating $P(v_i)$ is straightforward with a large training set; count the fraction of the set that are of class $v_i$.
- However, estimating $P(a_1, a_2, ..., a_n | v_i)$ is difficult unless our training set is *very* large. We need to see every possible attribute combination many

# 16-36: Naive Bayes assumption

- Naive Bayes assumes that all attributes are conditionally independent of each other.
- In this case, $P(a_1, a_2, ..., a_n|v_i) = \Pi_i P(a_i|v_i)$.
- This can be estimated from the training data.
- The classifier then picks the class with the highest probability according to this equation.
- Interestingly, Naive Bayes performs well even in cases where the conditional independence assumption fails.

# 16-37: Example

- Recall your tennis-playing problem from the decision tree homework.
- We want to use the training data and a Naive Bayes classifier to classify the following instance:
- Outlook = Sunny, Temperature = Cool, Humidity = high, Wind = Strong.

# 16-38: Example

| Day | Outlook | Temperature | Humidity | Wind | PlayTennis |
|---|---|---|---|---|---|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

# 16-39: Example

- Our priors are:
  - $P(PlayTennis = yes) = 9/14 = 0.64$
  - $P(PlayTennis = no) = 5/14 = 0.36$
- We can estimate:
  - $P(wind = strong|PlayTennis = yes) = 3/9 = 0.33$
  - $P(wind = strong|PlayTennis = no) = 3/5 = 0.6$
  - $P(humidity = high|PlayTennis = yes) = 3/9 = 0.33$
  - $P(humidity = high|PlayTennis = no) = 4/5 = 0.8$
  - $P(outlook = sunny|PlayTennis = yes) = 2/9 = 0.22$
  - $P(outlook = sunny|PlayTennis = no) = 3/5 = 0.6$
  - $P(temp = cool|PlayTennis = yes) = 3/9 = 0.33$
  - $P(temp = cool|PlayTennis = no) = 1/5 = 0.2$

# 16-40: Example

- $v_{yes} = P(yes)P(sunny|yes)P(cool|yes)P(high|yes)P(strong|yes) = 0.005$
- $v_{no} = P(no)P(sunny|no)P(cool|no)P(high|no)P(strong|no) = 0.0206$
- So we see that not playing tennis is the maximum likelihood hypothesis.
- Further, by normalizing, we see that the classifier predicts a $\frac{0.0206}{0.005+0.0206} = 0.80$ probability of not playing tennis.

# 16-41: Estimating Probabilities

- As we can see from this example, estimating probabilities through frequency is risky when our data set is small.
- We only have 5 negative examples, so we may not have an accurate estimate.
- A better approach is to use the following formula, called an $m$-estimate:
- $\frac{n_c+mp}{n+m}$
- Where $n_c$ is the number of individuals with the characteristic of interest (say Wind = strong), $n$ is the total number of positive/negative examples, $p$ is our prior estimate, and $m$ is a constant called the *equivalent sample size*.

# 16-42: Estimating Probabilities

- $m$ determines how heavily to weight $p$.
- $p$ is assumed to be uniform.
- So, in the Tennis example,
  $P(wind = strong|playTennis = no) = \frac{3+0.2m}{5+m}$
- We'll determine an $m$ based on sample size.
  - If $m$ is zero, we just use observed data.
  - If $m >> n$, we use the prior.
  - Otherwise $m$ lets us weight these parameters' relative influence.

# 16-43: Naive Bayes: Classify Spam

- One are where Naive Bayes has been very successful is in text classification.
  - Despite the violation of independence assumptions.
- Classifying spam is just a special case of text classification.
- Problem - given some emails labled ham or spam, determine the category of new and unseen documents.
- Our features will be the tokens that appear in a document.
- Based on this, we'll predict a category.

# 16-44: Classifying spam

- Naive Bayes is only one possible way to classify spam.
  - Rule-based systems (SpamAssassin)
  - Examining headers (broken From or Content-Type)
  - Blacklist/Whitelist
  - Challenge/response

# 16-45: Naive Bayes: Classify Spam

- Naive Bayes has several properties that make it nice as a spam classifier:
  - We don't need to encode specific rules
  - We can adapt as the types of spam change
  - Somewhat robust to spammers adding in extra text

# 16-46: Naive Bayes: Classify Spam

- For a given email, we'll want to compute the MAP hypothesis - that is, is:
  - $P(spam|t1, t2, .., tn)$ greater than
  - $P(ham|t1, t2, .., tn)$
- We can use Bayes' rule to rewrite these as:
  - $\alpha P(t1, t2, ...., tn|spam)P(spam)$
  - $\alpha P(t1, t2, ...., tn|ham)P(ham)$

# 16-47: Naive Bayes: Classify Spam

- We can then use the Naive Bayes assumption to rewrite these as:
  - $\alpha P(t1|spam)P(t2|spam)...P(tn|spam)P(spam)$
  - $\alpha P(t1|ham)P(t2|ham)...P(tn|ham)P(ham)$
- And this we know how to compute.

# 16-48: Naive Bayes: Classify Spam

- We can get the conditional probabilities by counting tokens in the training set.
- We can get the priors from the training set, or through estimation.

# 16-49: Naive Bayes: Classify Spam

- This is a case of a problem where we can tolerate occasional false negatives (spam classified as ham) but we cannot tolerate false positives (ham classified as spam).
- Plain old vanilla Naive Bayes will do fairly well, but there's a lot of tuning and tweaking that can be done to optimize performance.

# 16-50: Naive Bayes: Classify Spam

- There are a lot of wrinkles to consider:
  - What should be treated as a token? All words? All strings? Only some words?
  - Should headers be given different treatment? Greater or less emphasis? What about subject?
  - What about HTML?

# 16-51: Naive Bayes: Classify Spam

- There are a lot of wrinkles to consider:
  - When classifying an email, should you consider all tokens, or just the most significant?
  - When computing conditional probabilities, should you could the fraction of documents a token appear in, or the fraction of words represented by a particular token?
  - Can you use any of the NLTK tools to better identify structure?