Where do the probabilities come from?

- Probabilities come from:
  - Experts
  - Data
Learning probabilities — the simplest case

Observe tosses of thumbtack:

\( n_0 \) instances of \( Heads = false \)
\( n_1 \) instances of \( Heads = true \)

what should we use as \( P(heads) \)?
Observe tosses of thumbtack:

- $n_0$ instances of $Heads = false$
- $n_1$ instances of $Heads = true$

What should we use as $P(heads)$?

- Empirical frequency: $P(heads) = \frac{n_1}{n_0 + n_1}$
Learning probabilities — the simplest case

Observe tosses of thumbtack:

- $n_0$ instances of Heads = false
- $n_1$ instances of Heads = true

What should we use as $P(\text{heads})$?

- **Empirical frequency:** $P(\text{heads}) = \frac{n_1}{n_0 + n_1}$

- **Laplace smoothing [1812]:** $P(\text{heads}) = \frac{n_1 + 1}{n_0 + n_1 + 2}$

---

©D. Poole and A. Mackworth 2019  Artificial Intelligence, Lecture 10.1  2 / 6
Learning probabilities — the simplest case

Observe tosses of thumbtack:
- \( n_0 \) instances of *Heads* = false
- \( n_1 \) instances of *Heads* = true

What should we use as \( P(\text{heads}) \)?

- Empirical frequency: \( P(\text{heads}) = \frac{n_1}{n_0 + n_1} \)
- Laplace smoothing [1812]: \( P(\text{heads}) = \frac{n_1 + 1}{n_0 + n_1 + 2} \)
- Informed priors: \( P(\text{heads}) = \frac{n_1 + c_1}{n_0 + n_1 + c_0 + c_1} \)

for some informed pseudo counts \( c_0, c_1 > 0 \).

- \( c_0 = 1, c_1 = 1 \), expressed ignorance (uniform prior)

Pseudo-counts convey prior knowledge. Consider: “how much more would I believe \( \alpha \) if I had seen one example with \( \alpha \) true than if I has seen no examples with \( \alpha \) true?”
Observe tosses of thumbtack:

\( n_0 \) instances of \( Heads = false \)

\( n_1 \) instances of \( Heads = true \)

what should we use as \( P(\text{heads}) \)?

- **Empirical frequency:**
  \[
P(\text{heads}) = \frac{n_1}{n_0 + n_1}
\]

- **Laplace smoothing [1812]:**
  \[
P(\text{heads}) = \frac{n_1 + 1}{n_0 + n_1 + 2}
\]

- **Informed priors:**
  \[
P(\text{heads}) = \frac{n_1 + c_1}{n_0 + n_1 + c_0 + c_1}
\]

  for some informed pseudo counts \( c_0, c_1 > 0 \).

  \( c_0 = 1, c_1 = 1 \), expressed ignorance (uniform prior)

Pseudo-counts convey prior knowledge. Consider: “how much more would I believe \( \alpha \) if I had seen one example with \( \alpha \) true than if I has seen no examples with \( \alpha \) true?” — empirical frequency overfits to the data.
We have a web site where people rate restaurants with 1 to 5 stars.

We want to report the most liked restaurant(s) — the one predicted to have the best future ratings.

How can we determine the most liked restaurant?
We have a web site where people rate restaurants with 1 to 5 stars.

We want to report the most liked restaurant(s) — the one predicted to have the best future ratings.

How can we determine the most liked restaurant?

Are the restaurants with the highest average rating the most liked restaurants?
Example of Overfitting

- We have a web site where people rate restaurants with 1 to 5 stars.
- We want to report the most liked restaurant(s) — the one predicted to have the best future ratings.
- How can we determine the most liked restaurant?
- Are the restaurants with the highest average rating the most liked restaurants?
- Which restaurants have the highest average rating?
We have a web site where people rate restaurants with 1 to 5 stars.

We want to report the most liked restaurant(s) — the one predicted to have the best future ratings.

How can we determine the most liked restaurant?

Are the restaurants with the highest average rating the most liked restaurants?

Which restaurants have the highest average rating?

Which restaurants have a rating of 5?
We have a web site where people rate restaurants with 1 to 5 stars.

We want to report the most liked restaurant(s) — the one predicted to have the best future ratings.

How can we determine the most liked restaurant?

Are the restaurants with the highest average rating the most liked restaurants?

Which restaurants have the highest average rating?

Which restaurants have a rating of 5?

Solution: add some “average” ratings for each restaurant!
Probability of Heads

Toss 1  Toss 2  ...  Toss 11

aispace: http://artint.info/code/aispace/beta.xml

- Probability of Heads is a random variable representing the probability of heads.
- Range is \( \{0.0, 0.1, 0.2, \ldots, 0.9, 1.0\} \) or interval \([0, 1]\).
- \( P(Toss\#n=Heads \mid Probability\_of\_Heads=v) = \)
Probability of Heads

Toss 1  Toss 2  ...  Toss 11

aispace: http://artint.info/code/aispace/beta.xml

- *Probability of Heads* is a random variable representing the probability of heads.
- Range is \{0.0, 0.1, 0.2, \ldots, 0.9, 1.0\} or interval \([0, 1]\).
- \(P(Toss\#n=Heads \mid Probability\ of\ Heads=v) = v\)
- Toss\#i is independent of Toss\#j (for \(i \neq j\)) given Probability of Heads
- i.i.d. or independent and identically distributed.
$H$ is the help page the user is interested in. We observe the words in the query.
Naive Bayes Classifier: User’s request for help

\( H \) is the help page the user is interested in. We observe the words in the query. What probabilities are required?
$H$ is the help page the user is interested in. We observe the words in the query. What probabilities are required? What counts are required?
Naive Bayes Classifier: User’s request for help

$H$ is the help page the user is interested in. We observe the words in the query. What probabilities are required? What counts are required?

- number of times each help page $h_i$ is the best one
- number of times word $w_j$ is used when $h_i$ is the help page.
$H$ is the help page the user is interested in. We observe the words in the query. What probabilities are required? What counts are required?

- number of times each help page $h_i$ is the best one
- number of times word $w_j$ is used when $h_i$ is the help page.

When can the counts be updated?
$H$ is the help page the user is interested in. We observe the words in the query. What probabilities are required? What counts are required?
- number of times each help page $h_i$ is the best one
- number of times word $w_j$ is used when $h_i$ is the help page.
When can the counts be updated?
- When the correct page is found.
Naive Bayes Classifier: User’s request for help

$H$ is the help page the user is interested in. We observe the words in the query. What probabilities are required? What counts are required?

- number of times each help page $h_i$ is the best one
- number of times word $w_j$ is used when $h_i$ is the help page.

When can the counts be updated?
- When the correct page is found.

What prior counts should be used? Can they be zero?
If you were designing such a system, many issues arise such as:

- What if the most likely page isn’t the correct page?
If you were designing such a system, many issues arise such as:

- What if the most likely page isn’t the correct page?
- What if the user can’t find the correct page?
If you were designing such a system, many issues arise such as:

- What if the most likely page isn’t the correct page?
- What if the user can’t find the correct page?
- What if the user mistakenly thinks they have the correct page?
Issues

If you were designing such a system, many issues arise such as:

- What if the most likely page isn’t the correct page?
- What if the user can’t find the correct page?
- What if the user mistakenly thinks they have the correct page?
- Can some pages never be found?
If you were designing such a system, many issues arise such as:

- What if the most likely page isn’t the correct page?
- What if the user can’t find the correct page?
- What if the user mistakenly thinks they have the correct page?
- Can some pages never be found?
- What about common words?
If you were designing such a system, many issues arise such as:

- What if the most likely page isn’t the correct page?
- What if the user can’t find the correct page?
- What if the user mistakenly thinks they have the correct page?
- Can some pages never be found?
- What about common words?
- What about words that affect other words, e.g. “not”? 
If you were designing such a system, many issues arise such as:

- What if the most likely page isn’t the correct page?
- What if the user can’t find the correct page?
- What if the user mistakenly thinks they have the correct page?
- Can some pages never be found?
- What about common words?
- What about words that affect other words, e.g. “not”?  
- What about new words?
If you were designing such a system, many issues arise such as:

- What if the most likely page isn’t the correct page?
- What if the user can’t find the correct page?
- What if the user mistakenly thinks they have the correct page?
- Can some pages never be found?
- What about common words?
- What about words that affect other words, e.g. “not”? 
- What about new words?
- What do we do with new help pages?
If you were designing such a system, many issues arise such as:

- What if the most likely page isn’t the correct page?
- What if the user can’t find the correct page?
- What if the user mistakenly thinks they have the correct page?
- Can some pages never be found?
- What about common words?
- What about words that affect other words, e.g. “not”?  
- What about new words?
- What do we do with new help pages?
- How can we transfer the language model to a new help system?