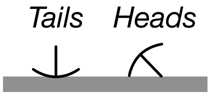
Where do the probabilities come from?

- Probabilities come from:
  - Experts
  - Data

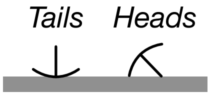
Learning Probabilities

Observe tosses of thumbtack:
n_0 instances of $Heads = false$
n_1 instances of $Heads = true$
what should we use as $P(heads)$?



Learning Probabilities

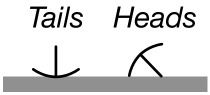
Observe tosses of thumbtack:
n_0 instances of $Heads = false$
n_1 instances of $Heads = true$
what should we use as $P(heads)$?



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

Learning Probabilities

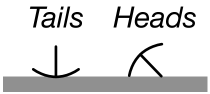
Observe tosses of thumbtack:
n_0 instances of $Heads = false$
n_1 instances of $Heads = true$
what should we use as $P(heads)$?



- Empirical frequency: $P(heads) = \dfrac{n_1}{n_0 + n_1}$
- Laplace smoothing [1812]: $P(heads) = \dfrac{n_1 + 1}{n_0 + n_1 + 2}$

Learning Probabilities

Observe tosses of thumbtack:
n_0 instances of $Heads = false$
n_1 instances of $Heads = true$
what should we use as $P(heads)$?

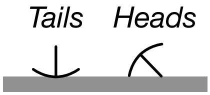


- Empirical frequency: $P(heads) = \dfrac{n_1}{n_0 + n_1}$
- Laplace smoothing [1812]: $P(heads) = \dfrac{n_1 + 1}{n_0 + n_1 + 2}$
- Informed priors: $P(heads) = \dfrac{n_1 + c_1}{n_0 + n_1 + c_0 + c_1}$
  for some informed pseudo counts $c_0, c_1 > 0$.
  $c_0 = 1$, $c_1 = 1$, expressed ignorance (uniform prior)

Pseudo-counts convey prior knowledge. Consider: "how much more would I believe α if I had seen one example with α true than if I has seen no examples with α true?"

Learning Probabilities

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: $P(heads) = \dfrac{n_1}{n_0 + n_1}$
- Laplace smoothing [1812]: $P(heads) = \dfrac{n_1 + 1}{n_0 + n_1 + 2}$
- Informed priors: $P(heads) = \dfrac{n_1 + c_1}{n_0 + n_1 + c_0 + c_1}$
  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 α if I had seen one example with α true than if I has seen no examples with α true?"
— empirical frequency overfits to the data.

Example of Ovefitting

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

Example of Ovefitting

- We have a web site where people rate restaurants with 1 to 5 stars.
- We want to report the most liked restaurant(s).
- How can we determine the most liked restaurant?
- Is the restuarant with the higest average rating the most liked restaurant?

Example of Ovefitting

- We have a web site where people rate restaurants with 1 to 5 stars.
- We want to report the most liked restaurant(s).
- How can we determine the most liked restaurant?
- Is the restuarant with the higest average rating the most liked restaurant?
- Which restaurants have the higest average rating?

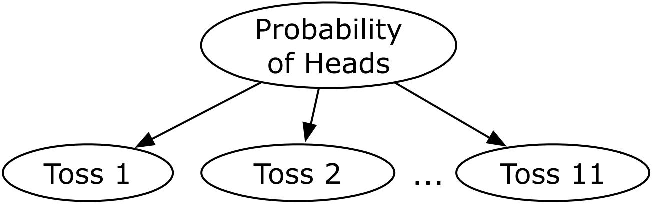
Example of Ovefitting

- We have a web site where people rate restaurants with 1 to 5 stars.
- We want to report the most liked restaurant(s).
- How can we determine the most liked restaurant?
- Is the restuarant with the higest average rating the most liked restaurant?
- Which restaurants have the higest average rating?
- Which restaurants have a rating of 5?

Example of Ovefitting

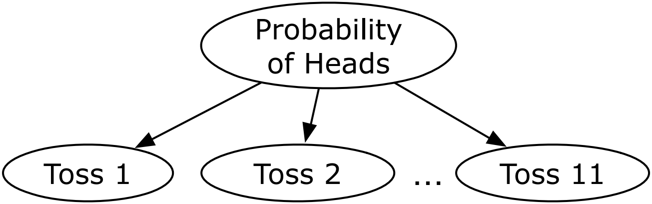
- We have a web site where people rate restaurants with 1 to 5 stars.
- We want to report the most liked restaurant(s).
- How can we determine the most liked restaurant?
- Is the restuarant with the higest average rating the most liked restaurant?
- Which restaurants have the higest average rating?
- Which restaurants have a rating of 5?
- Solution: add some “average” ratings for each restaurant!

Bayesian Learning



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

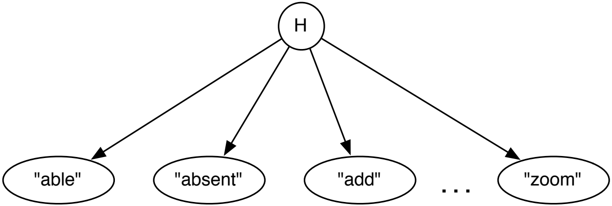
- $Probablity\_of\_Heads$ is a random variable representing the probability of heads.
- Range is $\{0.0, 0.1, 0.2, \ldots, 0.9, 1.0\}$ or interval $[0, 1]$.
- $P(Toss\#n{=}Heads \mid Probablity\_of\_Heads{=}v) =$



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

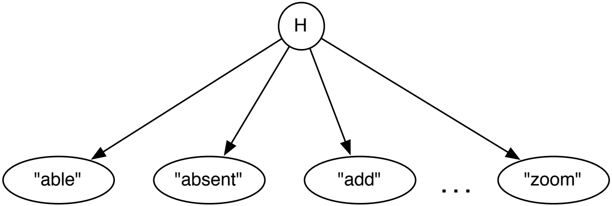
- *Probablity_of_Heads* is a random variable representing the probability of heads.
- Range is $\{0.0, 0.1, 0.2, \ldots, 0.9, 1.0\}$ or interval $[0, 1]$.
- $P(Toss\#n{=}Heads \mid Probablity\_of\_Heads{=}v) = v$
- $Toss\#i$ is independent of $Toss\#j$ (for $i \neq j$) given *Probablity_of_Heads*
- i.i.d. or independent and identically distributed.

Naive Bayes Classifier: User's request for help



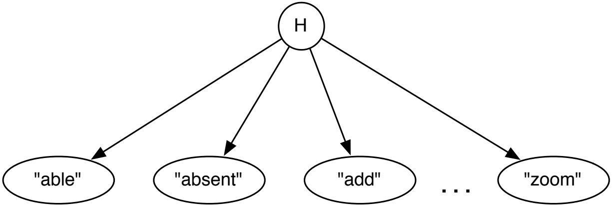
H is the help page the user is interested in.
We observe the words in the query.

Naive Bayes Classifier: User's request for help



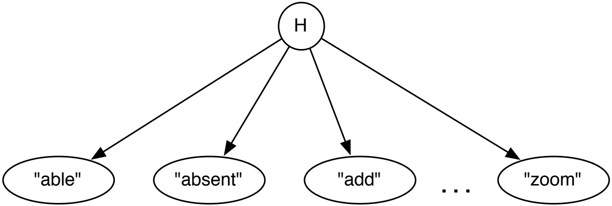
H is the help page the user is interested in.
We observe the words in the query.
What probabilities are required?

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?

Naive Bayes Classifier: User's request for help



H is the help page the user is interested in.

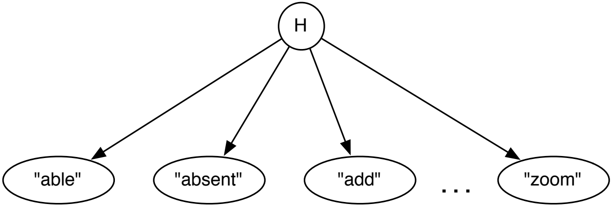
We observe the words in the query.

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.

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.

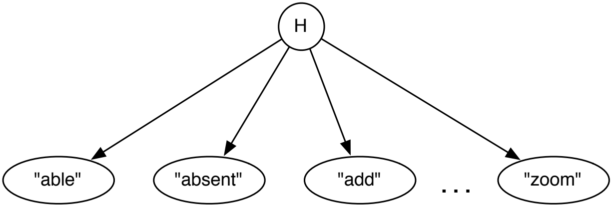
What probabilities are required?

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?

Naive Bayes Classifier: User's request for help



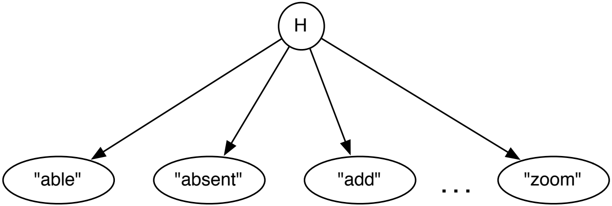
H is the help page the user is interested in.
We observe the words in the query.
What probabilities are required?
What counts are required?

- 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.

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?

Issues

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

- What if the most likely page isn't 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?

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?

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?

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?
- What about common words?

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?
- What about common words?
- What about words that affect other words, e.g. "not"?

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?
- What about common words?
- What about words that affect other words, e.g. "not"?
- What about new words?

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?
- 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?

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?
- 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?