Lecture 2: Probability, Naive Bayes

CS 585, Fall 2015
Introduction to Natural Language Processing
http://people.cs.umass.edu/~brenocon/inlp2015/

Brendan O'Connor

Today

- Probability Review
- “Naive Bayes” classification
- Python demo

Probability Theory Review

$$= \sum_a P(A=a)$$

Conditional Probability $$= \frac{P(AB)}{P(B)}$$

Chain Rule $$= P(A|B)P(B)$$

Law of Total Probability
$$= \sum_b P(A, B=b)$$
$$= \sum_b P(A|B=b)P(B=b)$$

Disjunction (Union) $$P(A \vee B) =$$

Negation (Complement) $$P(\neg A) =$$

Probability Theory Review

$$1 = \sum_a P(A = a)$$

Conditional Probability $\qquad = \frac{P(AB)}{P(B)}$

Chain Rule $\qquad = P(A|B)P(B)$

Law of Total Probability

$$= \sum_b P(A, B = b)$$

$$= \sum_b P(A|B = b)P(B = b)$$

Disjunction (Union) $\qquad P(A \vee B) =$

Negation (Complement) $\qquad P(\neg A) =$

Probability Theory Review

$$1 = \sum_a P(A = a)$$

Conditional Probability $$P(A|B) = \frac{P(AB)}{P(B)}$$

Chain Rule $$= P(A|B)P(B)$$

Law of Total Probability
$$= \sum_b P(A, B = b)$$
$$= \sum_b P(A|B = b)P(B = b)$$

Disjunction (Union) $$P(A \vee B) =$$

Negation (Complement) $$P(\neg A) =$$

Probability Theory Review

$$1 = \sum_a P(A = a)$$

Conditional Probability $$P(A|B) = \frac{P(AB)}{P(B)}$$

Chain Rule $$P(AB) = P(A|B)P(B)$$

Law of Total Probability
$$= \sum_b P(A, B = b)$$
$$= \sum_b P(A|B = b)P(B = b)$$

Disjunction (Union) $$P(A \vee B) =$$

Negation (Complement) $$P(\neg A) =$$

Probability Theory Review

$$1 = \sum_a P(A = a)$$

Conditional Probability
$$P(A|B) = \frac{P(AB)}{P(B)}$$

Chain Rule
$$P(AB) = P(A|B)P(B)$$

Law of Total Probability
$$P(A) = \sum_b P(A, B = b)$$
$$= \sum_b P(A|B = b)P(B = b)$$

Disjunction (Union)
$$P(A \vee B) =$$

Negation (Complement)
$$P(\neg A) =$$

Probability Theory Review

$$1 = \sum_a P(A = a)$$

Conditional Probability $$P(A|B) = \frac{P(AB)}{P(B)}$$

Chain Rule $$P(AB) = P(A|B)P(B)$$

Law of Total Probability
$$P(A) = \sum_b P(A, B = b)$$
$$P(A) = \sum_b P(A|B = b)P(B = b)$$

Disjunction (Union) $$P(A \vee B) =$$

Negation (Complement) $$P(\neg A) =$$

Probability Theory Review

$$1 = \sum_a P(A = a)$$

Conditional Probability $$P(A|B) = \frac{P(AB)}{P(B)}$$

Chain Rule $$P(AB) = P(A|B)P(B)$$

Law of Total Probability
$$P(A) = \sum_b P(A, B = b)$$
$$P(A) = \sum_b P(A|B = b)P(B = b)$$

Disjunction (Union) $$P(A \vee B) = P(A) + P(B) - P(AB)$$

Negation (Complement) $$P(\neg A) =$$

Probability Theory Review

$$1 = \sum_a P(A = a)$$

Conditional Probability $$P(A|B) = \frac{P(AB)}{P(B)}$$

Chain Rule $$P(AB) = P(A|B)P(B)$$

Law of Total Probability

$$P(A) = \sum_b P(A, B = b)$$

$$P(A) = \sum_b P(A|B = b)P(B = b)$$

Disjunction (Union) $$P(A \vee B) = P(A) + P(B) - P(AB)$$

Negation (Complement) $$P(\neg A) = 1 - P(A)$$

Bayes Rule

P(D|H)

Model: how hypothesis causes data

H: unknown → D: observed data / evidence

Bayesian inference

P(H|D)

Bayes Rule tells you how to flip the conditional.
Useful if you assume a *generative process* for your data.

Likelihood — Prior

$$P(H|D) = \frac{P(D|H)P(H)}{P(D)}$$

Posterior — Normalizer

Rev. Thomas Bayes
c. 1701-1761

Bayes Rule and its pesky denominator

Likelihood Prior

$$P(h|d) = \frac{P(d|h)P(h)}{P(d)} = \frac{P(d|h)P(h)}{\sum_{h'} P(d|h')P(h')}$$

Constant w.r.t. *h*

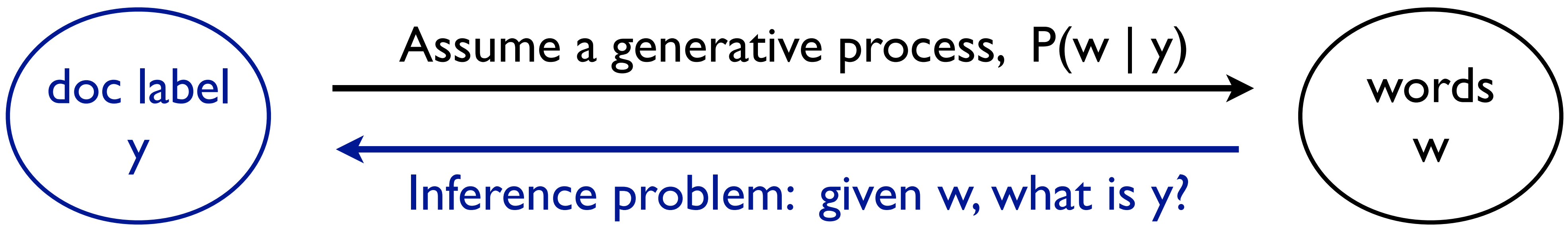
$$P(h|d) \propto \underline{P(d|h)P(h)}$$

Unnormalized posterior
By itself does not sum to 1!

$\propto$ "Proportional to"

Implicitly for varying H.
This notation is very common, though slightly ambiguous.

Bayes Rule for classification inference



Authorship problem: classify a new text.
Is it y=Anna or y=Barry?

Observe w: Look at random word in the new text.
It is *abracadabra.*

$$P(y=A \mid w=\textit{abracadabra}) \quad ?$$

$$P(y \mid w) = P(w|y) \; P(y) \; / \; P(w)$$

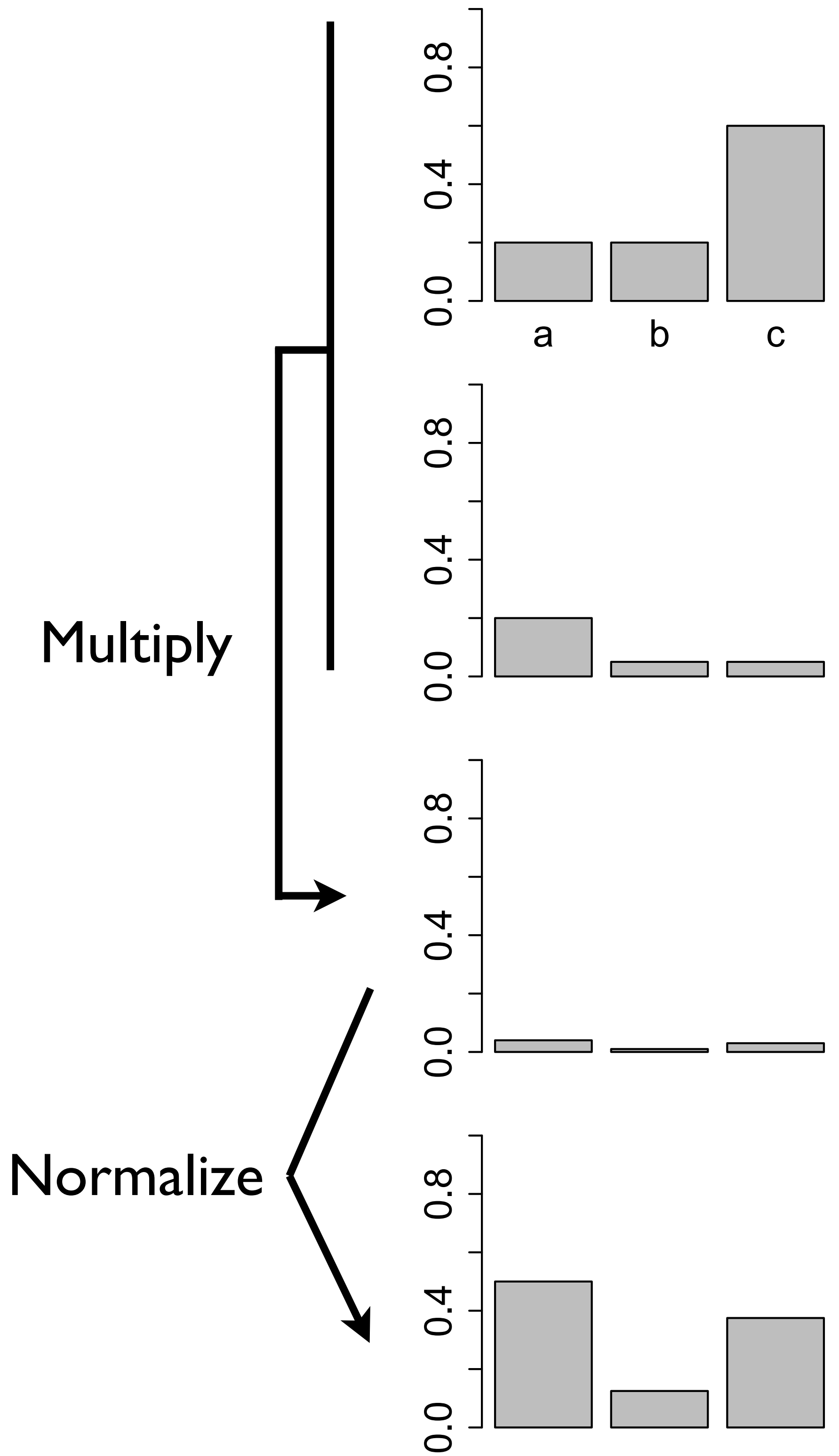
P(y): Assume 50% prior prob.

P(w | y):
Calculate from previous data

	abracadabra	*gesundheit*
Anna	5 per 1000 words	6 per 1000 words
Barry	10 per 1000 words	1 per 1000 words

Bayes Rule as hypothesis vector scoring

<u>Sum to 1?</u>



$$P(H = h)$$

Prior

$$P(E|H = h)$$

Likelihood

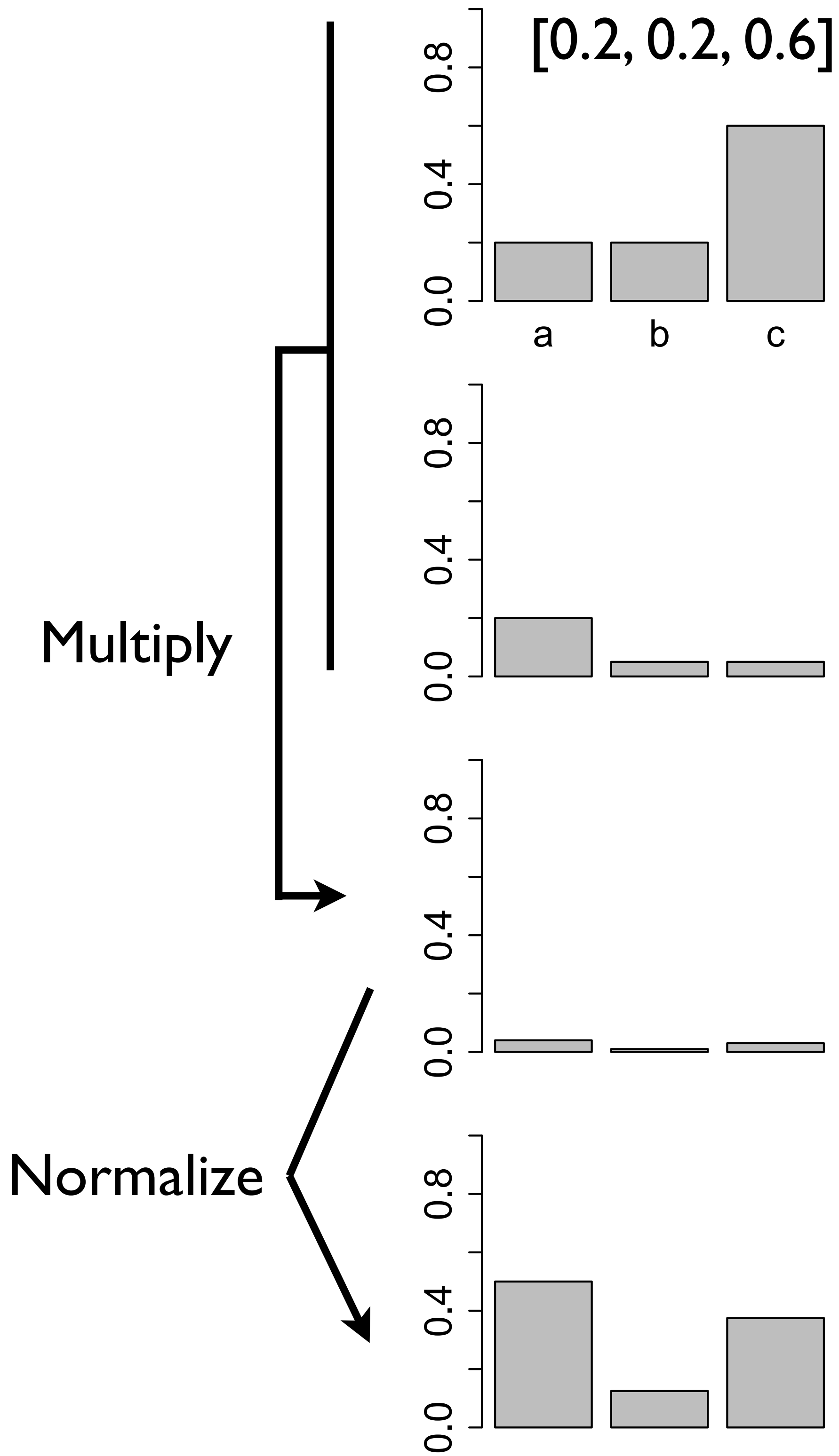
$$P(E|H = h)P(H = h)$$

Unnorm. Posterior

$$\frac{1}{Z}P(E|H = h)P(H = h)$$

Posterior

Bayes Rule as hypothesis vector scoring



Sum to 1?

$$P(H=h)$$

Prior

Yes

$$P(E|H=h)$$

Likelihood

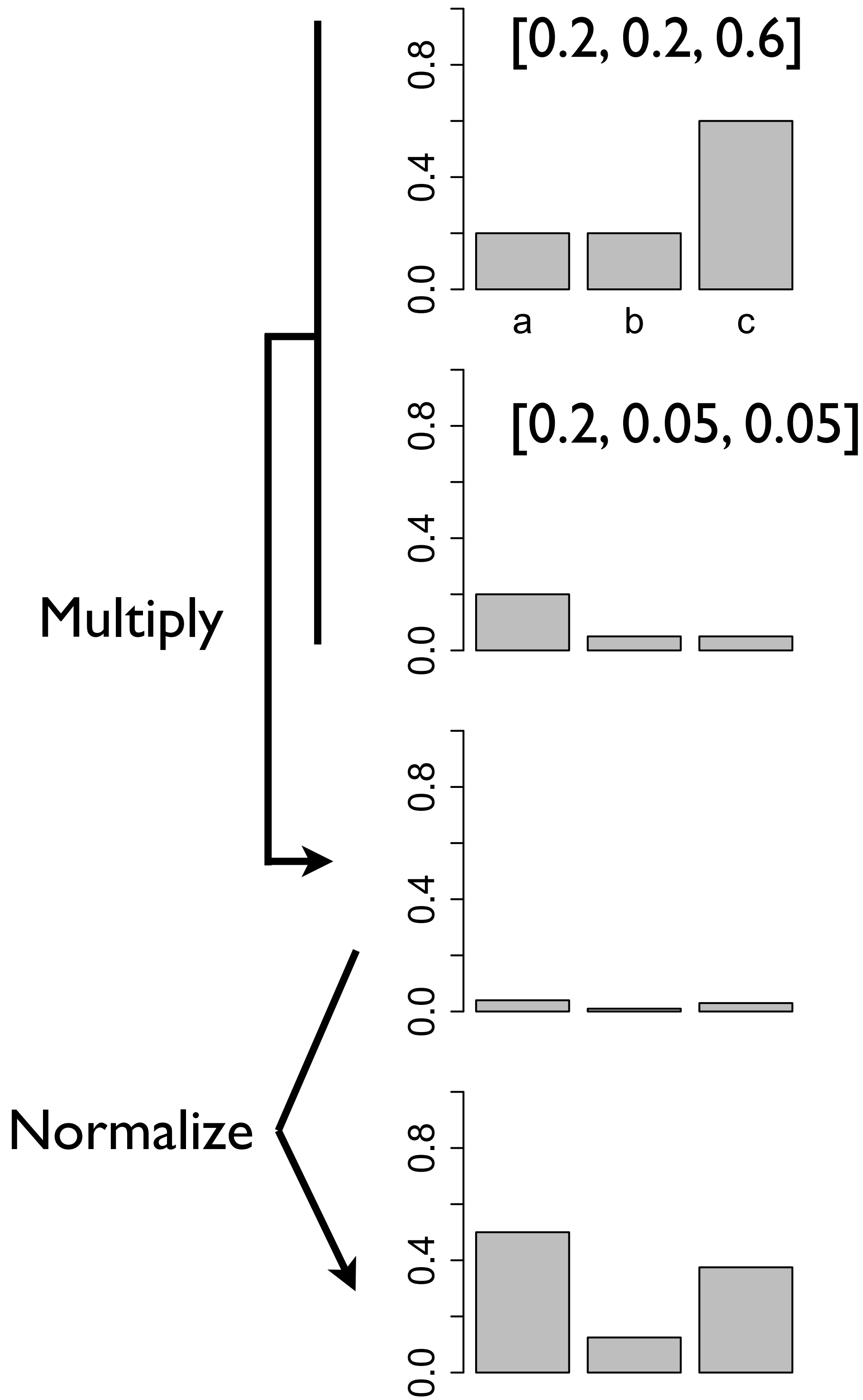
$$P(E|H=h)P(H=h)$$

Unnorm. Posterior

$$\frac{1}{Z}P(E|H=h)P(H=h)$$

Posterior

Bayes Rule as hypothesis vector scoring



Sum to 1?

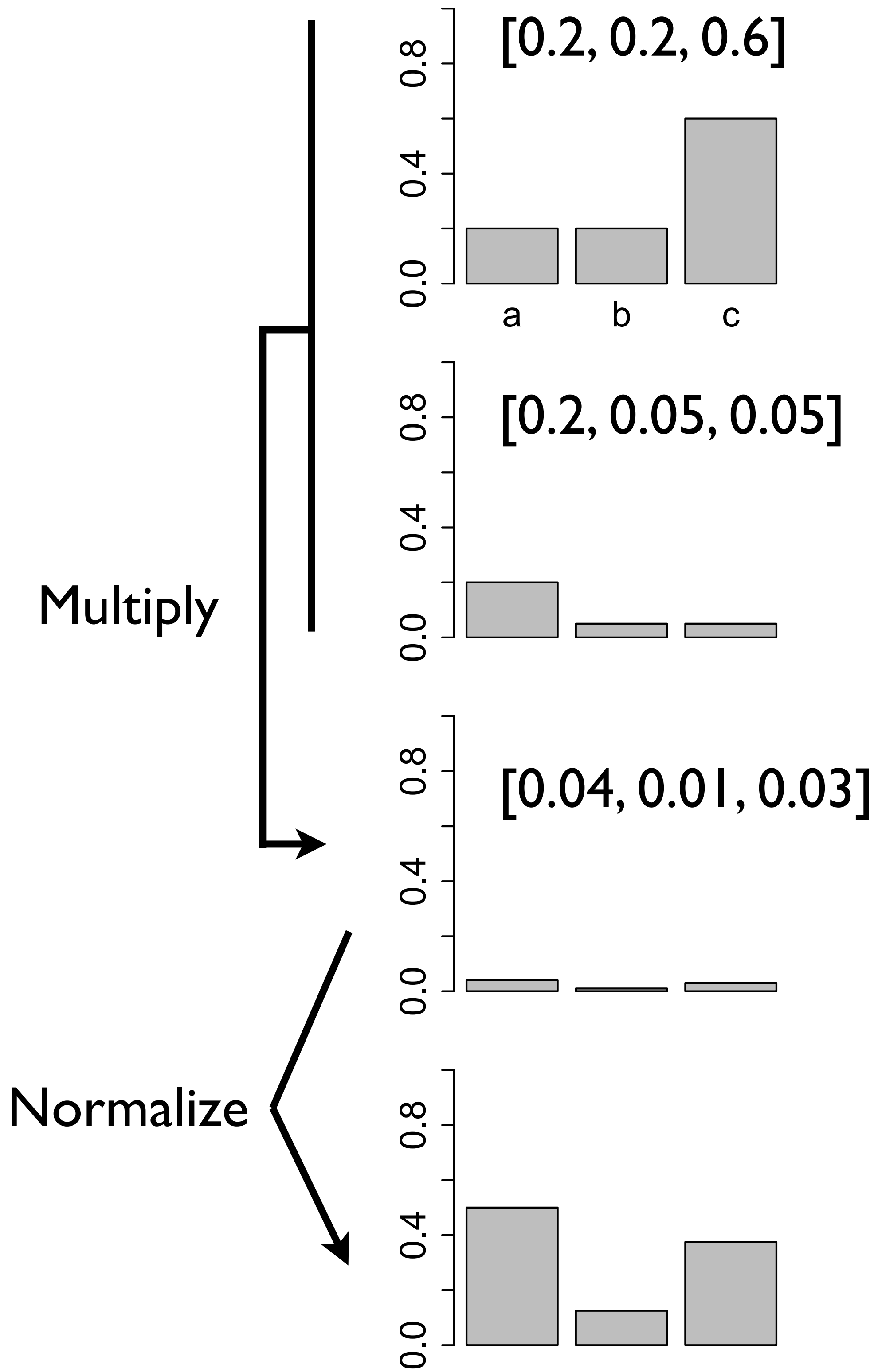
$P(H = h)$
Prior — Yes

$P(E|H = h)$
Likelihood — No

$P(E|H = h)P(H = h)$
Unnorm. Posterior

$\frac{1}{Z}P(E|H = h)P(H = h)$
Posterior

Bayes Rule as hypothesis vector scoring



Sum to 1?

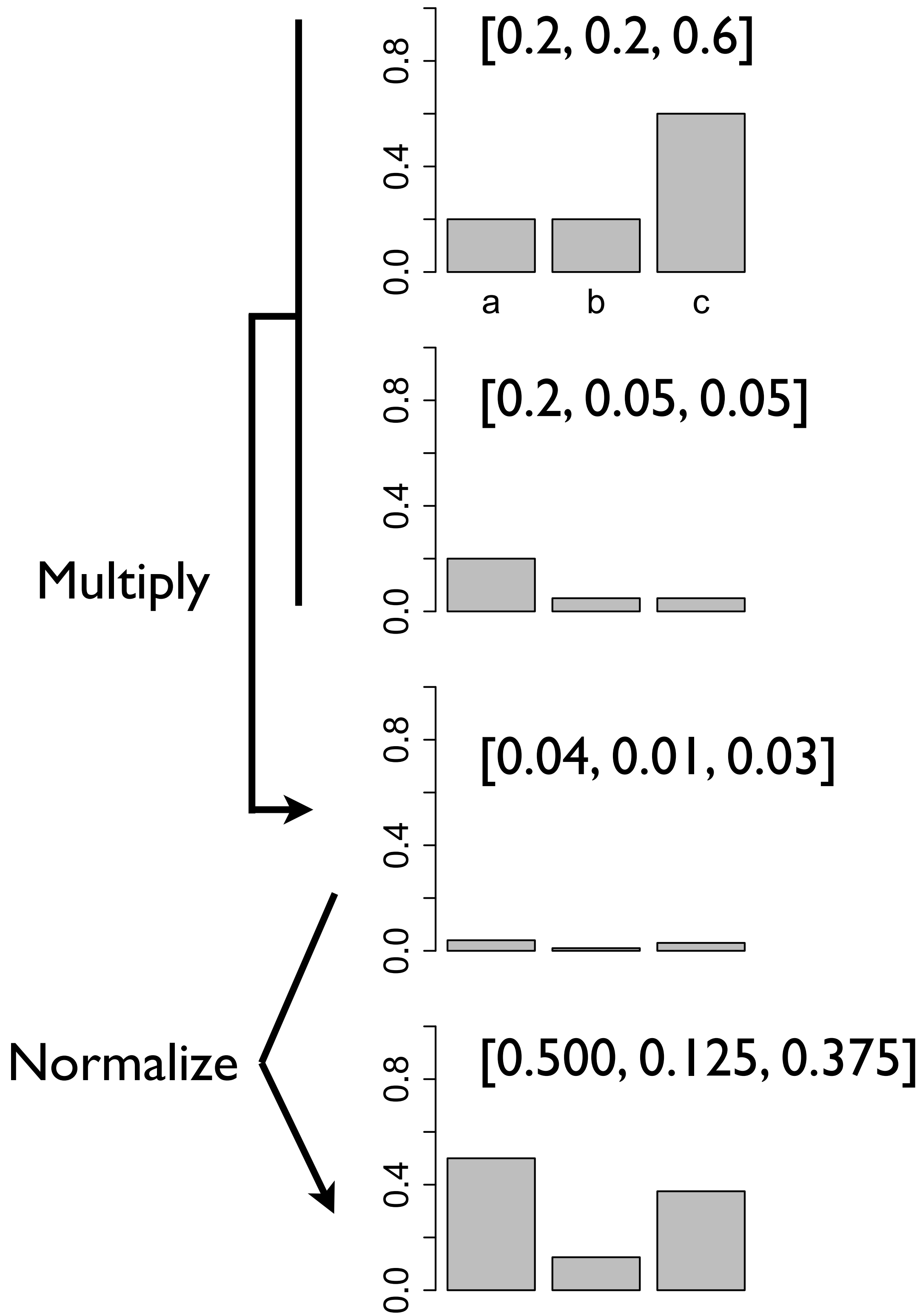
$P(H=h)$
Prior — Yes

$P(E|H=h)$
Likelihood — No

$P(E|H=h)P(H=h)$
Unnorm. Posterior — No

$$\frac{1}{Z}P(E|H=h)P(H=h)$$
Posterior

Bayes Rule as hypothesis vector scoring



	Sum to 1?
$P(H=h)$ Prior	Yes
$P(E\|H=h)$ Likelihood	No
$P(E\|H=h)P(H=h)$ Unnorm. Posterior	No
$\frac{1}{Z}P(E\|H=h)P(H=h)$ Posterior	Yes

Text Classification with Naive Bayes

Classification problems

- Given text ***d***, want to predict label ***y***
  - Is this restaurant review positive or negative?
  - Is this email spam or not?
  - Which author wrote this text?
  - (Is this word a noun or verb?)
- ***d***: documents, sentences, etc.
- ***y***: discrete/categorical variable

Goal: from training set of (d,y) pairs, *learn* a probabilistic classifier f(d) = P(y|d) ("supervised learning")

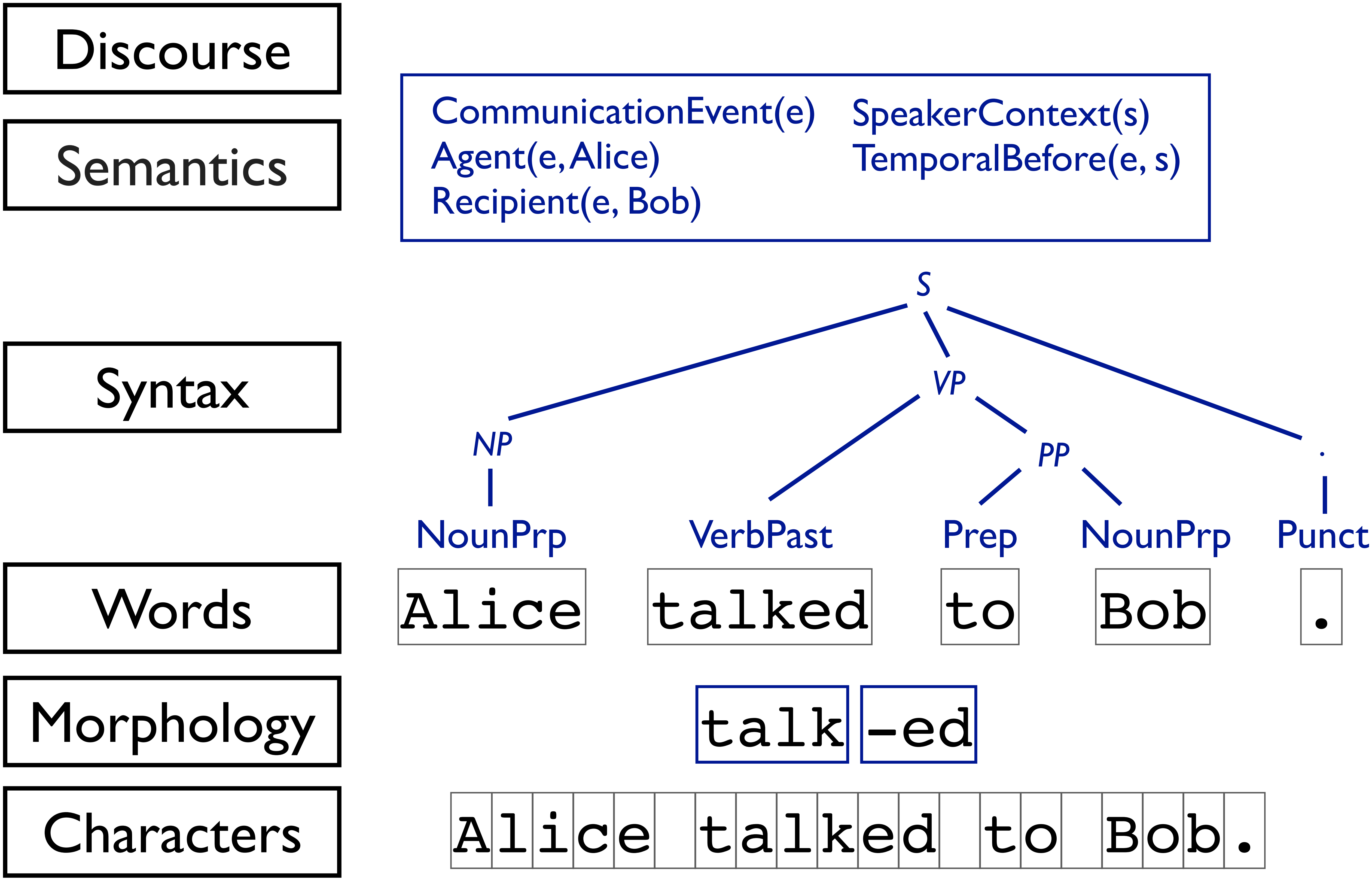
Features for model: Bag-of-words

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

Figure 6.1 Intuition of the multinomial naive Bayes classifier applied to a movie review. The position of the words is ignored (the *bag of words* assumption) and we make use of the frequency of each word.

Levels of linguistic structure



Levels of linguistic structure

Words	Alice	talked	to	Bob	.

Characters `Alice talked to Bob.`

Levels of linguistic structure

Words are fundamental units of meaning

Words	`Alice`	`talked`	`to`	`Bob`	`.`

Characters	`A`	`l`	`i`	`c`	`e`	` `	`t`	`a`	`l`	`k`	`e`	`d`	` `	`t`	`o`	` `	`B`	`o`	`b`	`.`

Levels of linguistic structure

Words are fundamental units of meaning

and easily identifiable*

*in some languages

Words	`Alice` `talked` `to` `Bob` `.`
Characters	`Alice talked to Bob.`

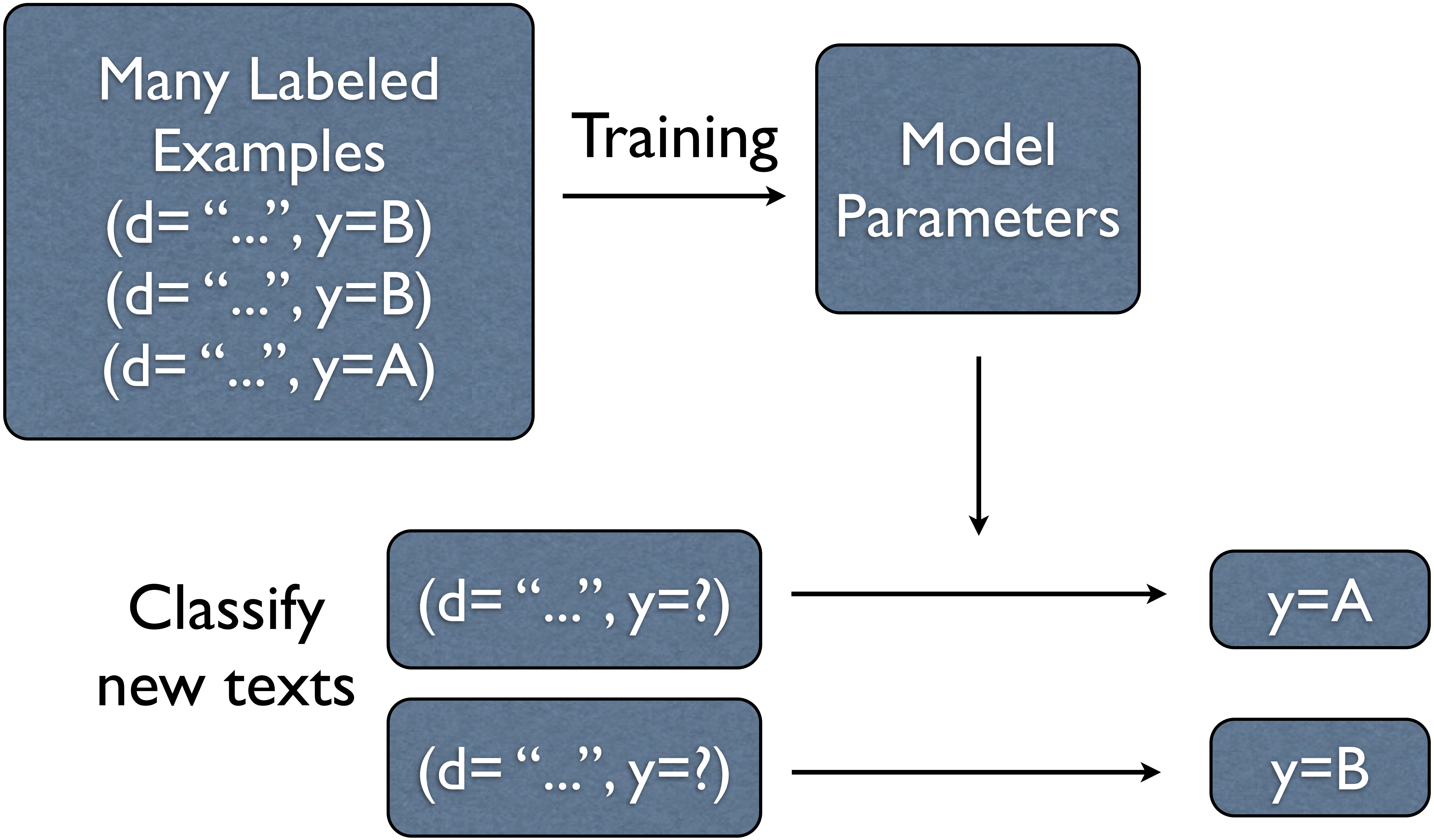
How to classify with words?

- Approach #1: use a predefined dictionary (or make one up)
  Human Knowledge
  - e.g. for sentiment....
    - score += 1 for each “happy”, “awesome”, “cool”
    - score -= 1 for each “sad”, “awful”, “bad”

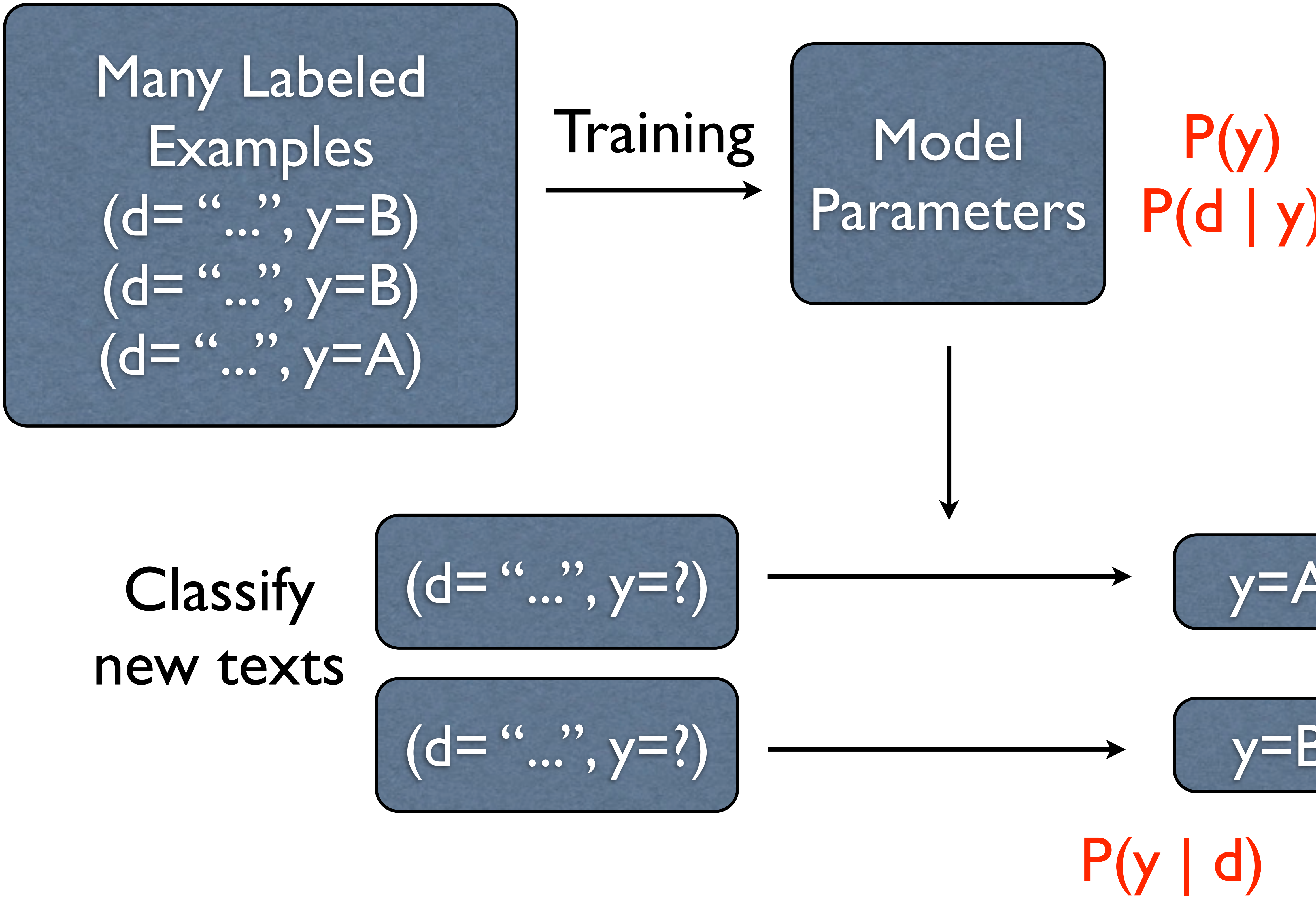
How to classify with words?

- Approach #1: use a predefined dictionary (or make one up)
  Human Knowledge
  - e.g. for sentiment....
    - score += 1 for each "happy", "awesome", "cool"
    - score -= 1 for each "sad", "awful", "bad"
- Approach #2: use labeled documents
  Supervised Learning
  - Learn which words correlate to positive vs. negative documents
  - Use these correlations to classify new documents

Supervised learning



Supervised learning: Generative model



Multinomial Naive Bayes

$$P(y \mid w_1..w_T) \propto P(y)\ P(w_1..w_T \mid y)$$

Tokens in doc

Predictions:

Predict class $\arg\max_y P(Y = y \mid w_1..w_T)$

or, predict prob of classes...

Multinomial Naive Bayes

$$P(y \mid w_1..w_T) \propto P(y)\ P(w_1..w_T \mid y)$$

Tokens in doc

$$\prod_t P(w_t \mid y)$$

the "Naive Bayes" assumption: conditional indep.

Parameters: $P(w \mid y)$ for each document category ***y*** and wordtype ***w***

$P(y)$ prior distribution over document categories ***y***

Learning: Estimate parameters as frequency ratios; e.g.

$$P(w \mid y, \alpha) = \frac{\#(w \text{ occurrences in docs with label } y) + \alpha}{\#(\text{tokens total across docs with label } y) + V\alpha}$$

Predictions:

Predict class $\arg\max_y P(Y = y \mid w_1..w_T)$

or, predict prob of classes...