Natural Language Processing

Text classification

Min Jang

yuliats@cs.washington.edu

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

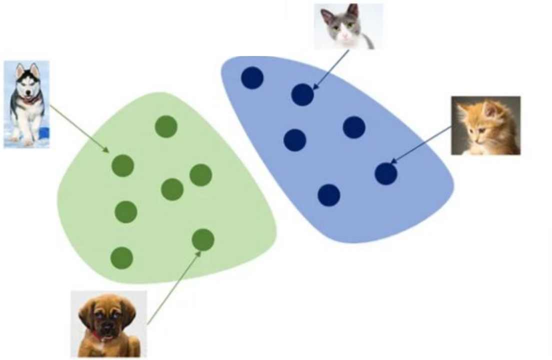
Announcements

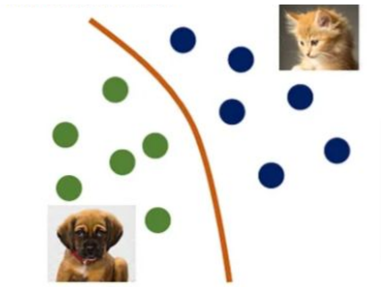
- Quiz 1 – good luck!

Generative vs discriminative models

- Learns the input distribution
- Maximizes the joint probability: P(X, Y)
- Estimates P(X|Y) to find P(Y|X) using Bayes' rule
- Can generate new data
- Typically, they are NOT used to solve classification tasks
- Generative models possess discriminative properties

Hidden Markov Models | Naive Bayes | Gaussian Mixture Models | Gaussian Discriminant Analysis | LDA | Bayesian Networks





- Learns the decision boundary between classes
- Maximizes the conditional probability: P(Y|X)
- Directly estimates P(Y|X)
- Cannot generate new data
- Specifically meant for classification tasks
- Discriminative models don't possess generative properties

Logistic Regression | Random Forests | SVMs | Neural Networks | Decision Tree | kNN

https://blog.dailydoseofds.com/p/an-intuitive-guide-to-generative
https://medium.com/@jordi299/about-generative-and-discriminative-models-d8958b67ad32

Generative and discriminative models

- Generative text classification: Learn a model of the joint P(X, y), and find

$$\hat{y} = \underset{\tilde{y}}{\operatorname{argmax}}\ P(X, \tilde{y})$$

- Discriminative text classification: Learn a model of the conditional P(y | X), and find

$$\hat{y} = \underset{\tilde{y}}{\operatorname{argmax}}\ P(\tilde{y} | X)$$

Finding the correct class c from a document d in Generative vs Discriminative Classifiers

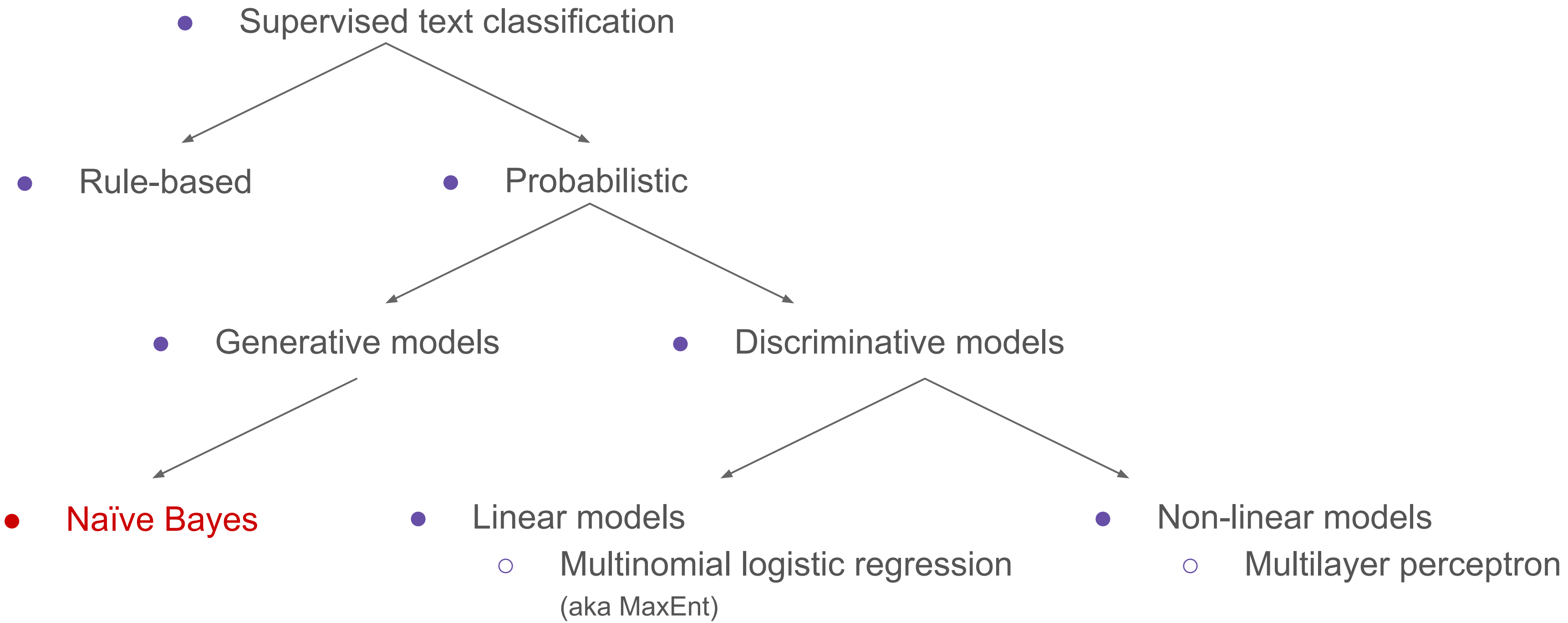
- Naive Bayes

$$\hat{c} = \operatorname*{argmax}_{c \in C} \ \overbrace{P(d|c)}^{\text{likelihood}} \ \overbrace{P(c)}^{\text{prior}}$$

- Logistic Regression

$$\hat{c} = \operatorname*{argmax}_{c \in C} \ \overset{\text{posterior}}{P(c|d)}$$

We’ll consider alternative models for classification



- Supervised text classification
- Rule-based
- Probabilistic
- Generative models
- Discriminative models
- Naïve Bayes
- Linear models
  - Multinomial logistic regression (aka MaxEnt)
- Non-linear models
  - Multilayer perceptron

Generative text classification: naïve Bayes

- Simple classification method
  - based on the Bayes rule
- Relies on very simple (naïve) representation of a documents
  - Conditional independence assumption: the features are conditionally independent, given the target class (hence the name "naïve")
  - bag-of-words, no relative order
- A good baseline for more sophisticated models

Andrew Y. Ng and Michael I. Jordan, On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes, Advances in Neural Information Processing Systems 14 (NIPS), 2001.

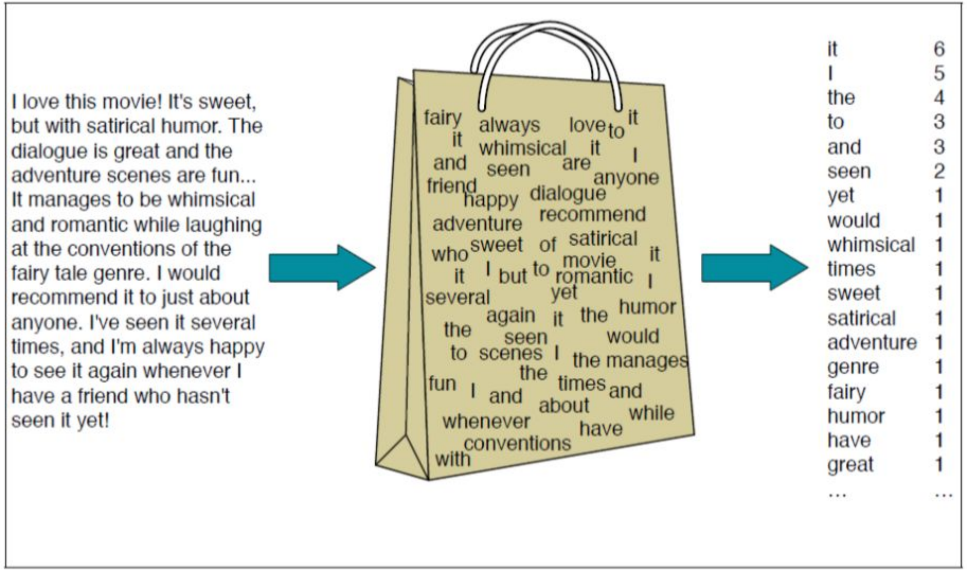
Naïve Bayes

Sentiment analysis: movie reviews

- Given a document d (e.g., a movie review)
- Decide which class c it belongs to: positive, negative, neutral
- Compute $P(c \mid d)$ for each c
  - $P(\text{positive} \mid d)$, $P(\text{negative} \mid d)$, $P(\text{neutral} \mid d)$
  - select the one with max P

Bag-of-Words (BOW)

- Given a document d (e.g., a movie review) – how to represent d ?



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 7.1 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.

Figure from J&M 3rd ed. draft, sec 7.1

Naïve Bayes

- Given a document d and a class c, use Bayes' rule:

$$\underbrace{P(c|d)}_{\text{posterior}} = \frac{P(d|c)P(c)}{P(d)}$$

Naïve Bayes

- Given a document d and a class c, Bayes' rule:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

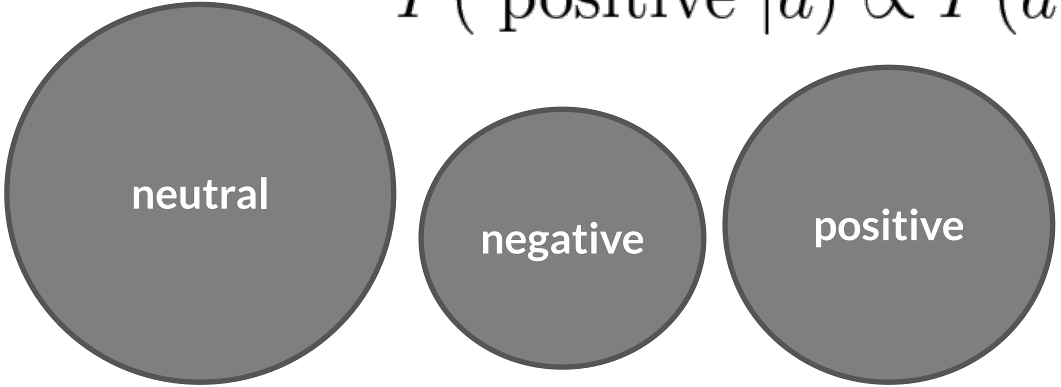
$$\underset{\substack{\Big\downarrow \\ \text{posterior}}}{P(\text{‘positive’}|d)} \propto \underset{\substack{\Big\downarrow \\ \text{likelihood}}}{P(d|\text{‘positive’})}\,\underset{\substack{\Big\downarrow \\ \text{prior}}}{P(\text{‘positive’})}$$

Naïve Bayes

- Given a document d and a class c, Bayes' rule:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

$$P(\text{`positive'}|d) \propto P(d|\text{`positive'})\underbrace{P(\text{`positive'})}_{\text{prior}}$$



Naïve Bayes

- Given a document d and a class c, Bayes' rule:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

$$P(\text{‘positive’}|d) \propto \underset{\text{likelihood}}{\underset{\big\downarrow}{P(d|\text{‘positive’})}}P(\text{‘positive’})$$

Naïve Bayes independence assumptions

$$P(w_1, w_2, \ldots, w_n | c)$$

- **Bag of Words assumption**: Assume position doesn't matter
- **Conditional Independence**: Assume the feature probabilities $P(w_i | c_j)$ are independent given the class c

$$P(w_1, w_2, \ldots, w_n | c) = P(w_1|c) \times P(w_2|c) \times P(w_3|c) \times \ldots \times P(w_n|c)$$

Document representation

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!

→ **bag of words (BOW)** →

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
…	…

Document representation

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!

→ **bag of words (BOW)** →

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
…	…

$$P(d|c) = P(w_1, w_2, \ldots, w_n|c) = \prod_i P(w_i|c)$$

Generative text classification: Naïve Bayes

$$\mathrm{C}_{NB} = \underset{c}{\operatorname{argmax}} P(c|d) = \underset{c}{\operatorname{argmax}} \frac{P(d|c)P(c)}{P(d)} \propto$$

Bayes rule

$$\underset{c}{\operatorname{argmax}} P(d|c)P(c) =$$

same denominator

$$\underset{c}{\operatorname{argmax}} P(w_1, w_2, \ldots, w_n|c)P(c) =$$

representation

$$\underset{c_j}{\operatorname{argmax}} \; P(c_j) \prod_i P(w_i|c)$$

conditional independence

Underflow prevention: log space

- Multiplying lots of probabilities can result in floating-point underflow
- Since log(xy) = log(x) + log(y)
  - better to sum logs of probabilities instead of multiplying probabilities
- Class with highest un-normalized log probability score is still most probable

$$\mathrm{C}_{NB} = \underset{c_j}{\operatorname{argmax}}\, P(c_j) \prod_i P(w_i|c)$$

$$\mathrm{C}_{NB} = \underset{c_j}{\operatorname{argmax}}\, log(P(c_j)) + \sum_i log(P(w_i|c))$$

- Model is now just max of sum of weights

Learning the multinomial naïve Bayes

- How do we learn (train) the NB model?

Learning the multinomial naïve Bayes

- How do we learn (train) the NB model?
- We learn $P(c)$ and $P(w_i|c)$ from training (labeled) data

$$\mathrm{C}_{NB} = \operatorname*{argmax}_{c_j} log(\underline{P(c_j)}) + \textstyle\sum_i log(\underline{P(w_i|c)})$$

Parameter estimation for NB

- Parameter estimation during training
- Concatenate all documents with category c into one mega-document
- Use the frequency of w_i in the mega-document to estimate the word probability

$$\mathrm{C}_{NB} = \underset{c_j}{\operatorname{argmax}}\, log(P(c_j)) + \textstyle\sum_i log(P(w_i|c))$$

$$\hat{P}(c_j) = \frac{doccount(C = c_j)}{N_{doc}}$$

$$\hat{P}(w_i|c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

Parameter estimation for NB

$$\hat{P}(w_i|c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

- fraction of times word w_i appears among all words in documents of topic c_j

- Create mega-document for topic j by concatenating all docs in this topic
  - Use frequency of w in mega-document

Problem with Maximum Likelihood

- What if we have seen no training documents with the word “fantastic” and classified in the topic positive?

Problem with Maximum Likelihood

- What if we have seen no training documents with the word “fantastic” and classified in the topic positive?

$$\hat{P}(\text{“}fantastic\text{”}|c = \text{positive}) = \frac{count(\text{“}fantastic\text{”}, \text{positive})}{\sum_{w \in V} count(w, \text{positive})} = 0$$

- Zero probabilities cannot be conditioned away, no matter the other evidence!

$$\underset{c_j}{\text{argmax}}\; P(c_j) \prod_i P(w_i|c)$$

Laplace (add-1) smoothing for naïve Bayes

$$\hat{P}(w_i|c_j) = \frac{count(w_i, c_j) + 1}{\sum_{w \in V}(count(w, c_j) + 1)}$$

Laplace (add-1) smoothing for naïve Bayes

$$\hat{P}(w_i|c_j) = \frac{count(w_i, c_j) + 1}{\sum_{w \in V}(count(w, c_j) + 1)}$$

$$= \frac{count(w_i, c_j) + 1}{(\sum_{w \in V}(count(w, c_j)) + |V|}$$

- Note about log space

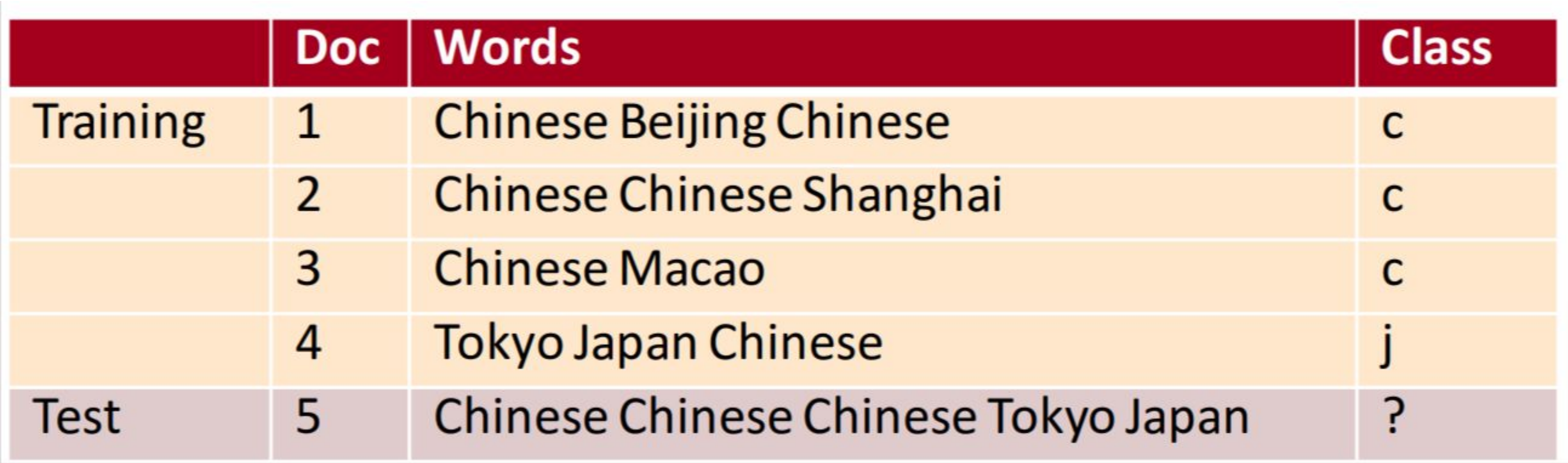
Multinomial naïve Bayes : learning

- From training corpus, extract *Vocabulary*
- Calculate $P(c_j)$ terms
  - For each c_j do
    - $docs_j \leftarrow$ all docs with class = c_j
    - $P(c_j) \leftarrow \frac{|docs_j|}{total\ \#\ documents}$

Multinomial naïve Bayes : learning

- From training corpus, extract *Vocabulary*
- Calculate $P(c_j)$ terms
  - For each c_j do
    - $docs_j \leftarrow$ all docs with class = c_j
    - $P(c_j) \leftarrow \dfrac{|docs_j|}{total\ \#\ documents}$
- Calculate $P(w_i| c_j)$ terms
  - $Text_j \leftarrow$ single doc containing all $docs_j$
  - For each word w_i in *Vocabulary*
    - $n_i \leftarrow$ # of occurrences of w_i in $Text_j$
    - $P(w_j | c_j) \leftarrow \dfrac{n_i + \alpha}{n + \alpha|Vocabulary|}$

Example



	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

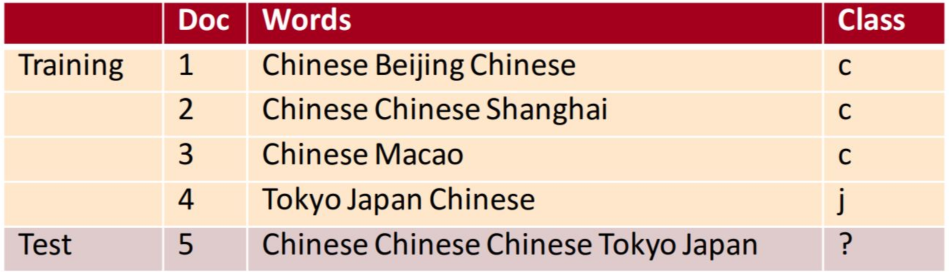
Example

$$\hat{P}(c) = \frac{N_c}{N}$$

Priors:

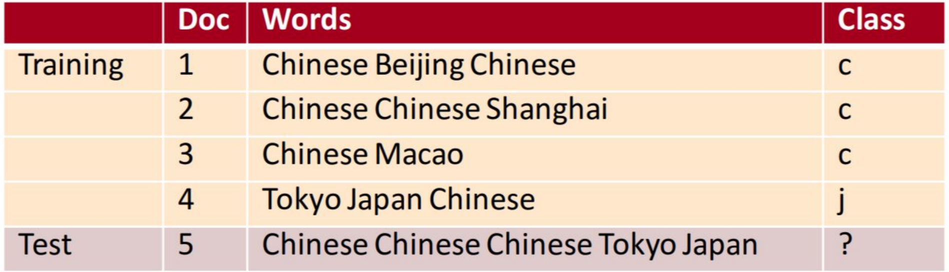
$P(c) = \frac{3}{4}$

$P(j) = \frac{1}{4}$



	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

Example



	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

Priors:

$P(c)= \frac{3}{4}$

$P(j)= \frac{1}{4}$

Conditional Probabilities:

P(Chinese|*c*) = (5+1) / (8+6) = 6/14 = 3/7

P(Tokyo|*c*) = (0+1) / (8+6) = 1/14

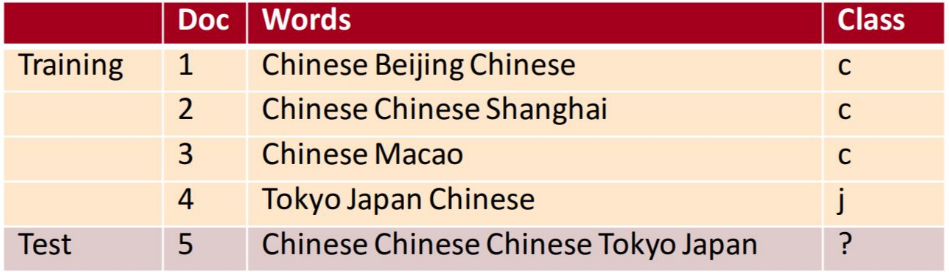
P(Japan|*c*) = (0+1) / (8+6) = 1/14

P(Chinese|*j*) = (1+1) / (3+6) = 2/9

P(Tokyo|*j*) = (1+1) / (3+6) = 2/9

P(Japan|*j*) = (1+1) / (3+6) = 2/9

Example



	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

Priors:

$P(c) = \frac{3}{4}$

$P(j) = \frac{1}{4}$

Conditional Probabilities:

P(Chinese|*c*) = (5+1) / (8+6) = 6/14 = 3/7

P(Tokyo|*c*) = (0+1) / (8+6) = 1/14

P(Japan|*c*) = (0+1) / (8+6) = 1/14

P(Chinese|*j*) = (1+1) / (3+6) = 2/9

P(Tokyo|*j*) = (1+1) / (3+6) = 2/9

P(Japan|*j*) = (1+1) / (3+6) = 2/9

Choosing a class:

$P(c|d5) \propto 3/4 * (3/7)^3 * 1/14 * 1/14$
≈ 0.0003

$P(j|d5) \propto 1/4 * (2/9)^3 * 2/9 * 2/9$
≈ 0.0001

Summary: naïve Bayes is not so naïve

- Naïve Bayes is a probabilistic model
- Naïve because is assumes features are independent of each other for a class
- Very fast, low storage requirements
- Robust to Irrelevant Features
  - Irrelevant Features cancel each other without affecting results
- Very good in domains with many equally important features
  - Decision Trees suffer from fragmentation in such cases – especially if little data
- Optimal if the independence assumptions hold: If assumed independence is correct, then it is the Bayes Optimal Classifier for problem
- A good dependable baseline for text classification
  - But we will see other classifiers that give better accuracy