# *Text Classification - II*

Pawan Goyal

CSE, IIT Kharagpur

Week 11, Lecture 5

# A worked example

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

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

| | 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 | ? |

# A worked example

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

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

| | 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 | ? |

**Priors:**

$P(c)=$

$P(j)=$

# A worked example

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

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

| | 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 | ? |

**Priors:**

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

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

**Conditional Probabilities:**

P(Chinese|*c*) =

P(Tokyo|*c*) =

P(Japan|*c*) =

P(Chinese|*j*) =

P(Tokyo|*j*) =

P(Japan|*j*) =

# A worked example

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

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

| | 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 | ? |

**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

# A worked example

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

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

| | 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 | ? |

**Priors:**

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

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

**Choosing a class:**

P(c|d5) $\propto$

P(j|d5) $\propto$

**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

# A worked example

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

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

| | 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 | ? |

**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:**

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

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

# Naïve Bayes and Language Modeling

*In general, NB classifier can use any feature*

URL, email addresses, dictionaries, network features

# *Naïve Bayes and Language Modeling*

*In general, NB classifier can use any feature*

URL, email addresses, dictionaries, network features

*But if we use only the word features and all the words in the text*

Naïve Bayes has an important similarity to language modeling.

# *Naïve Bayes and Language Modeling*

*In general, NB classifier can use any feature*

URL, email addresses, dictionaries, network features

*But if we use only the word features and all the words in the text*

Naïve Bayes has an important similarity to language modeling.

*Each class can be thought of as a separate unigram language model.*

# Naïve Bayes as Language Modeling

Which class assigns a higher probability to the sentence?

| Model pos | |
|---|---|
| 0.1 | I |
| 0.1 | love |
| 0.01 | this |
| 0.05 | fun |
| 0.1 | film |

| Model neg | |
|---|---|
| 0.2 | I |
| 0.001 | love |
| 0.01 | this |
| 0.005 | fun |
| 0.1 | film |

| I | love | this | fun | film |
|---|---|---|---|---|
| 0.1 | 0.1 | 0.01 | 0.05 | 0.1 |
| 0.2 | 0.001 | 0.01 | 0.005 | 0.1 |

$$P(s|pos) > P(s|neg)$$

# Naïve Bayes: More than Two Classes

### *Multi-value classification*

A document can belong to 0, 1 or $> 1$ classes

# *Naïve Bayes: More than Two Classes*

### *Multi-value classification*

A document can belong to 0, 1 or $> 1$ classes

### *Handling Multi-value classification*

- For each class $c \in C$, build a classifier $\gamma_c$ to distinguish $c$ from all other classes $c' \in C$

# Naïve Bayes: More than Two Classes

### *Multi-value classification*

A document can belong to 0, 1 or $> 1$ classes

### *Handling Multi-value classification*

- For each class $c \in C$, build a classifier $\gamma_c$ to distinguish $c$ from all other classes $c' \in C$
- Given test-doc $d$, evaluate it for membership in each class using each $\gamma_c$

# Naïve Bayes: More than Two Classes

### Multi-value classification

A document can belong to 0, 1 or $> 1$ classes

### Handling Multi-value classification

- For each class $c \in C$, build a classifier $\gamma_c$ to distinguish $c$ from all other classes $c' \in C$
- Given test-doc $d$, evaluate it for membership in each class using each $\gamma_c$
- $d$ belongs to any class for which $\gamma_c$ returns true

# *Naïve Bayes: More than Two Classes*

*One-of or multinomial classification*

Classes are mutually exclusive: each document in exactly one class

# *Naïve Bayes: More than Two Classes*

*One-of or multinomial classification*

Classes are mutually exclusive: each document in exactly one class

*Binary classifiers may also be used*

- For each class $c \in C$, build a classifier $\gamma_c$ to distinguish $c$ from all other classes $c' \in C$

# *Naïve Bayes: More than Two Classes*

### *One-of or multinomial classification*

Classes are mutually exclusive: each document in exactly one class

### *Binary classifiers may also be used*

- For each class $c \in C$, build a classifier $\gamma_c$ to distinguish $c$ from all other classes $c' \in C$
- Given test-doc $d$, evaluate it for membership in each class using each $\gamma_c$

# *Naïve Bayes: More than Two Classes*

*One-of or multinomial classification*

Classes are mutually exclusive: each document in exactly one class

*Binary classifiers may also be used*

- For each class $c \in C$, build a classifier $\gamma_c$ to distinguish $c$ from all other classes $c' \in C$
- Given test-doc $d$, evaluate it for membership in each class using each $\gamma_c$
- $d$ belongs to one class with maximum score

# Evaluation: Constructing Confusion matrix c

For each pair of classes $< c_1, c_2 >$ how many documents from $c_1$ were incorrectly assigned to $c_2$? (when $c_2 \neq c_1$)

| Docs in test set | Assigned UK | Assigned poultry | Assigned wheat | Assigned coffee | Assigned interest | Assigned trade |
|---|---|---|---|---|---|---|
| True UK | 95 | 1 | 13 | 0 | 1 | 0 |
| True poultry | 0 | 1 | 0 | 0 | 0 | 0 |
| True wheat | 10 | 90 | 0 | 1 | 0 | 0 |
| True coffee | 0 | 0 | 0 | 34 | 3 | 7 |
| True interest | - | 1 | 2 | 13 | 26 | 5 |
| True trade | 0 | 0 | 2 | 14 | 5 | 10 |

# Per class evaluation measures

## Recall

# Per class evaluation measures

## Recall

Fraction of docs in class $i$ classified correctly: $\frac{c_{ii}}{\sum_j c_{ij}}$

# *Per class evaluation measures*

### *Recall*

Fraction of docs in class $i$ classified correctly: $\frac{c_{ii}}{\sum_j c_{ij}}$

### *Precision*

Fraction of docs assigned class $i$ that are actually about class $i$:

# Per class evaluation measures

## Recall

Fraction of docs in class $i$ classified correctly: $\frac{c_{ii}}{\sum_j c_{ij}}$

## Precision

Fraction of docs assigned class $i$ that are actually about class $i$: $\frac{c_{ii}}{\sum_i c_{ji}}$

# Per class evaluation measures

### Recall

Fraction of docs in class $i$ classified correctly: $\frac{c_{ii}}{\sum\limits_j c_{ij}}$

### Precision

Fraction of docs assigned class $i$ that are actually about class $i$: $\frac{c_{ii}}{\sum\limits_i c_{ji}}$

### Accuracy

Fraction of docs classified correctly: $\frac{\sum\limits_i c_{ii}}{N}$

# Micro- vs. Macro-Average

If we have more than one class, how do we combine multiple performance measures into one quantity?

# *Micro- vs. Macro-Average*

If we have more than one class, how do we combine multiple performance measures into one quantity?

*Macro-averaging*

Compute performance for each class, then average

# *Micro- vs. Macro-Average*

If we have more than one class, how do we combine multiple performance measures into one quantity?

### *Macro-averaging*

Compute performance for each class, then average

### *Micro-averaging*

Collect decisions for all the classes, compute contingency table, evaluate.

# Micro- vs. Macro-Average

### Class 1

| | Truth: yes | Truth: no |
|---|---|---|
| Classifier: yes | 10 | 10 |
| Classifier: no | 10 | 970 |

### Class 2

| | Truth: yes | Truth: no |
|---|---|---|
| Classifier: yes | 90 | 10 |
| Classifier: no | 10 | 890 |

### Micro Ave. Table

| | Truth: yes | Truth: no |
|---|---|---|
| Classifier: yes | 100 | 20 |
| Classifier: no | 20 | 1860 |

# Micro- vs. Macro-Average

**Class 1**

| | Truth: yes | Truth: no |
|---|---|---|
| Classifier: yes | 10 | 10 |
| Classifier: no | 10 | 970 |

**Class 2**

| | Truth: yes | Truth: no |
|---|---|---|
| Classifier: yes | 90 | 10 |
| Classifier: no | 10 | 890 |

**Micro Ave. Table**

| | Truth: yes | Truth: no |
|---|---|---|
| Classifier: yes | 100 | 20 |
| Classifier: no | 20 | 1860 |

- Macro-averaged precision:

# Micro- vs. Macro-Average

**Class 1**

| | Truth: yes | Truth: no |
|---|---|---|
| Classifier: yes | 10 | 10 |
| Classifier: no | 10 | 970 |

**Class 2**

| | Truth: yes | Truth: no |
|---|---|---|
| Classifier: yes | 90 | 10 |
| Classifier: no | 10 | 890 |

**Micro Ave. Table**

| | Truth: yes | Truth: no |
|---|---|---|
| Classifier: yes | 100 | 20 |
| Classifier: no | 20 | 1860 |

- Macro-averaged precision: $(0.5 + 0.9)/2 = 0.7$
- Micro-averaged precision:

# Micro- vs. Macro-Average

**Class 1**

| | Truth: yes | Truth: no |
|---|---|---|
| Classifier: yes | 10 | 10 |
| Classifier: no | 10 | 970 |

**Class 2**

| | Truth: yes | Truth: no |
|---|---|---|
| Classifier: yes | 90 | 10 |
| Classifier: no | 10 | 890 |

**Micro Ave. Table**

| | Truth: yes | Truth: no |
|---|---|---|
| Classifier: yes | 100 | 20 |
| Classifier: no | 20 | 1860 |

- Macro-averaged precision: $(0.5 + 0.9)/2 = 0.7$
- Micro-averaged precision: $100/120 = 0.83$

# Micro- vs. Macro-Average

Class 1

| | Truth: yes | Truth: no |
|---|---|---|
| Classifier: yes | 10 | 10 |
| Classifier: no | 10 | 970 |

Class 2

| | Truth: yes | Truth: no |
|---|---|---|
| Classifier: yes | 90 | 10 |
| Classifier: no | 10 | 890 |

Micro Ave. Table

| | Truth: yes | Truth: no |
|---|---|---|
| Classifier: yes | 100 | 20 |
| Classifier: no | 20 | 1860 |

- Macro-averaged precision: $(0.5 + 0.9)/2 = 0.7$
- Micro-averaged precision: $100/120 = 0.83$

*Micro-averaged score is dominated by score on common classes*