

COMP9414: Artificial Intelligence

Lecture 6b: Text Classification

Wayne Wobcke

e-mail:w.wobcke@unsw.edu.au

This Lecture

- Probabilistic Formulation of Text Classification
- Rule-Based Text Classification
- Bayesian Text Classification
 - ▶ Bernoulli Model
 - ▶ Multinomial Naive Bayes
- Evaluating Classifiers

Text Classification Applications

- Spam Detection
- Authorship Analysis
- E-Mail Classification/Prioritization
- News/Scientific Article Topic Classification
- Event Extraction (Event Type Classification)
- Sentiment Analysis
- Recommender Systems (using Product Reviews)

Example Movie Reviews/Ratings

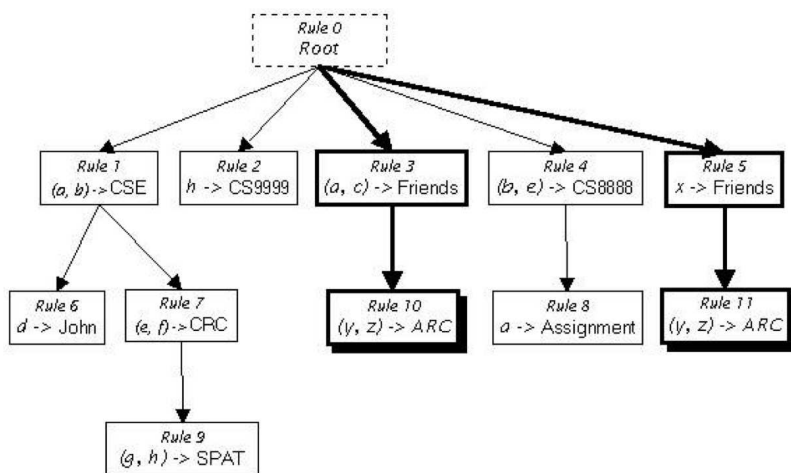
... unbelievably disappointing ...

Full of zany characters and richly applied satire, and some great plot twists.

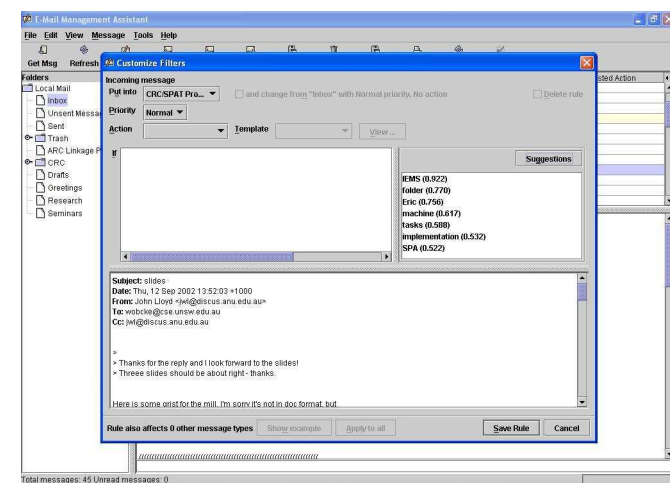
The greatest screwball comedy ever filmed.

It was pathetic. The worst part about it was the boxing scenes.

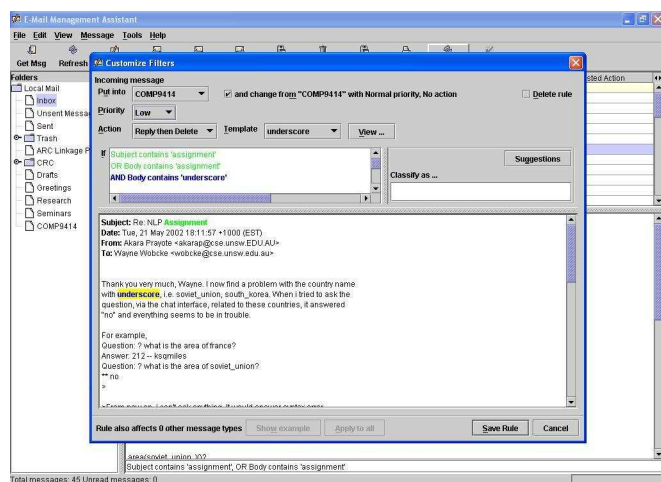
Rule-Based Method



Suggest Features using Naive Bayes



Help User Define Rules



Supervised Learning

- Input: A **document** (e-mail, news article, review, **tweet**)
- Output: One **class** drawn from a **fixed set** of classes
 - So text classification is a **multi-class** classification problem
 - ... and sometimes a **multi-label** classification problem
- Learning Problem
 - Input: Training set of labelled documents $\{(d_1, c_1), \dots\}$
 - Output: Learned classifier that maps d to predicted class c

Probabilistic Formulation

- Events: Occurrence of **features** x , occurrence of document of class c
- Given document x_1, \dots, x_n , choose c so that $P(c|x_1, \dots, x_n)$ is maximized
- Apply Bayes' Rule
 - ▶ $P(c|x_1, \dots, x_n) = \frac{P(x_1, \dots, x_n|c) \cdot P(c)}{P(x_1, \dots, x_n)}$
 - ▶ Therefore maximize $P(x_1, \dots, x_n|c) \cdot P(c)$

Bernoulli Model

Maximize $P(x_1, \dots, x_n|c) \cdot P(c)$

- Features are presence **or absence** of word w_i in document
- Apply independence assumptions
 - ▶ $P(x_1, \dots, x_n|c) = P(x_1|c) \cdot \dots \cdot P(x_n|c)$
 - ▶ Probability of word w (not) in class c independent of context
- Estimate probabilities
 - ▶ $P(w|c) = \#(w \text{ in document in class } c) / \#(\text{documents in class } c)$
 - ▶ $P(\neg w|c) = 1 - P(w|c)$
 - ▶ $P(c) = \#(\text{documents in class } c) / \#(\text{documents})$

Feature Engineering

Example SpamAssassin (Spam E-Mail)

- Mentions Generic Viagra
- Online Pharmacy
- Mentions millions of (dollar) ((dollar) NN,NNN,NNN.NN)
- Phrase: impress ... girl
- From: starts with many numbers
- Subject is all capitals
- HTML has a low ratio of text to image area
- One hundred percent guaranteed
- Claims you can be removed from the list

http://spamassassin.apache.org/old/tests_3_3_x.html

Naive Bayes Classification

w_1	w_2	w_3	w_4	Class
1	0	0	1	1
0	0	0	1	0
1	1	0	1	0
1	0	1	1	1
0	1	1	0	0
1	0	0	0	0
1	0	1	0	1
0	1	0	0	1
0	1	0	1	0
1	1	1	0	0

	Class = 1	Class = 0
$P(Class)$	0.40	0.60
$P(w_1 Class)$	0.75	0.50
$P(w_2 Class)$	0.25	0.67
$P(w_3 Class)$	0.50	0.33
$P(w_4 Class)$	0.50	0.50

To classify document with w_2, w_3, w_4

- $P(Class = 1 | \neg w_1, w_2, w_3, w_4)$
 $\approx ((1 - 0.75) * 0.25 * 0.5 * 0.5) * 0.4$
 $= 0.00625$
- $P(Class = 0 | \neg w_1, w_2, w_3, w_4)$
 $\approx ((1 - 0.5) * 0.5 * 0.67 * 0.33) * 0.6$
 $= 0.03333$

Bag of Words Model

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

Laplace Smoothing

- What if word in test document has not occurred in training?
- Then $P(w|c) = 0$ and so estimate for class c is 0
- Laplace smoothing
 - Assign small probability to unseen words
 - $P(w|c) = (\#(w \text{ in document } c) + 1) / (\sum_{w \in V} \#(w \text{ in document } c) + |V|)$
 - Don't have to add 1, can be 0.05 or some parameter α

Naive Bayes Classification

Maximize $P(x_1, \dots, x_n | c) \cdot P(c)$

- Features are occurrence of word in **positions** in document
- Apply independence assumptions
 - $P(w_1, \dots, w_n | c) = P(w_1 | c) \cdot \dots \cdot P(w_n | c)$
 - Position of word w in document doesn't matter
- Estimate probabilities
 - Let V be the vocabulary
 - Let "document" c = concatenation of documents in class c
 - $P(w|c) = \#(w \text{ in document } c) / \sum_{w \in V} \#(w \text{ in document } c)$
 - $P(c) = \#(\text{documents in class } c) / \#(\text{documents})$

MNB Example

	Words	Class
d_1	Chinese Beijing Chinese	c
d_2	Chinese Chinese Shanghai	c
d_3	Chinese Macao	c
d_4	Tokyo Japan Chinese	j
d_5	Chinese Chinese Chinese Tokyo Japan	?

$$P(\text{Chinese} | c) = (5+1)/(8+6) = 3/7$$

$$P(\text{Tokyo} | c) = (0+1)/(8+6) = 1/14$$

$$P(\text{Japan} | c) = (0+1)/(8+6) = 1/14$$

$$P(\text{Chinese} | j) = (1+1)/(3+6) = 2/9$$

$$P(\text{Tokyo} | j) = (1+1)/(3+6) = 2/9$$

$$P(\text{Japan} | j) = (1+1)/(3+6) = 2/9$$

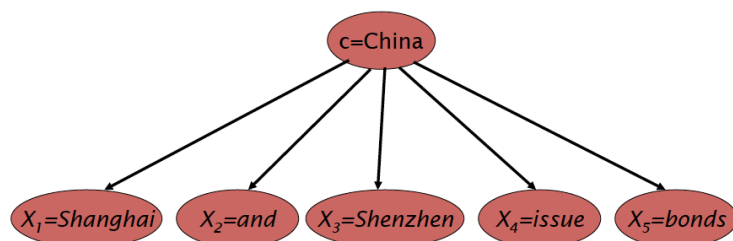
To classify document d_5

- $P(c | d_5) \propto [(3/7)^3 \cdot 1/14 \cdot 1/14] \cdot 3/4 \approx 0.0003$

- $P(j | d_5) \propto [(2/9)^3 \cdot 2/9 \cdot 2/9] \cdot 1/4 \approx 0.0001$

- Choose **Class c**

Graphical Model for Example



Multiple Classes: Per-Class Metrics

$n \times n$ Confusion Matrix (each instance in one class)

	Predicted c_1	Predicted c_2	...
Class c_1	c_{11}	c_{12}	c_{13}
Class c_2	c_{21}	c_{22}	c_{23}
...	c_{31}	c_{32}	c_{33}

- Precision (class c_i) = $c_{ii} / \sum_j c_{ji}$
 - ▶ Proportion of items predicted as c_i correctly classified (as c_i)
- Recall (class c_i) = $c_{ii} / \sum_j c_{ij}$
 - ▶ Proportion of items in class c_i predicted correctly (as c_i)
- Accuracy = $\sum_i c_{ii} / \sum_i \sum_j c_{ij}$

Evaluating Classifiers

2×2 Contingency Table (single class c)

	Class c	not Class c
Predicted c	True Positive	False Positive
Predicted not c	False Negative	True Negative

- Precision (P) = $TP / (TP + FP)$ – you want what you get
 - ▶ ... but may not get much
- Recall (R) = $TP / (TP + FN)$ – you get what you want
 - ▶ ... but you might get a lot more (junk)
- F1 = $2PR / (P + R)$ – harmonic mean of precision and recall

Multiple Classes: Micro/Macro-Averaging

n (one per class) 2×2 Contingency Tables

- Micro-average = Aggregated measure over all classes
 - ▶ micro-precision = $\sum_c TP_c / \sum_c (TP_c + FP_c)$
 - ▶ micro-recall = $\sum_c TP_c / \sum_c (TP_c + FN_c)$
 - ▶ Same when each instance has and is given one and only one label
 - ▶ Dominated by larger classes
- Macro-average = Average of per-class measures
 - ▶ macro-precision = $\frac{1}{n} \sum_c TP_c / (TP_c + FP_c)$
 - ▶ macro-recall = $\frac{1}{n} \sum_c TP_c / (TP_c + FN_c)$
 - ▶ Dominated by smaller classes
 - ▶ Fairer for imbalanced data, e.g. sentiment analysis

Summary: Naive Bayes

- 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
- Good dependable baseline for text classification