# COM6115: Text Processing

## *Sentiment Analysis: Approaches and Evaluation*

Chenghua Lin

Department of Computer Science
University of Sheffield

# Overview

- Definition of the problem of sentiment analysis
- **Approaches to sentiment analysis**
- **Evaluation of sentiment analysis approaches**

# Two approaches to SA

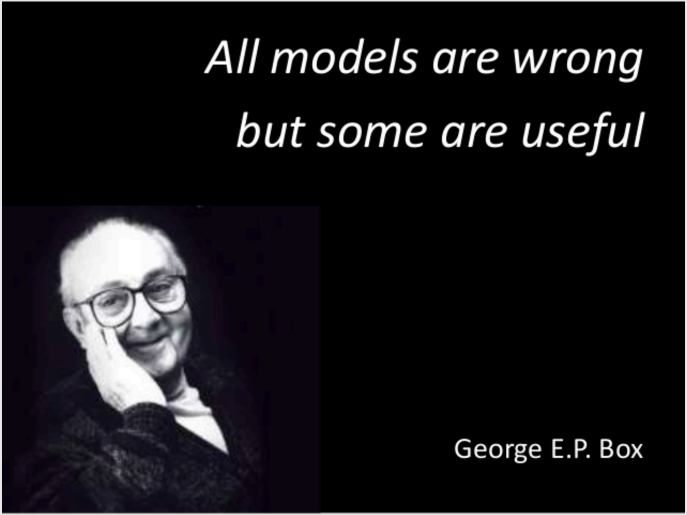
- Lexicon-based
  - Binary
  - Gradable
- **Corpus-based (machine learning)**

# Learning Outcomes

By the end of the SA sessions, you will be able to:

- Explain the relevance of the topic
- Differentiate between objective and subjective texts
- List the main elements in a sentiment analysis system
- Provide a critical summary of the main approaches for the problem
- Explain how sentiment analysis systems are evaluated.

# Text Processing



# Two Event Models for Naive Bayes

- Today we learn about Naïve Bayes classifier:
  - ◇ How to turn Bayes rule into a classifier
  - ◇ A supervised probabilistic model of the observed data
  - ◇ Can be used to predict the class label of new/unseen data
- Multi-variate Bernoulli Model: a document is a binary vector over the space of words
- Multinomial Model: captures word frequency information in documents

# Two Event Models for Naive Bayes

## A Comparison of Event Models for Naive Bayes Text Classification

**Andrew McCallum**[‡†]
mccallum@justresearch.com
[‡]Just Research
4616 Henry Street
Pittsburgh, PA 15213

**Kamal Nigam**[†]
knigam@cs.cmu.edu

[†]School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

**Abstract**

Recent approaches to text classification have used two different first-order probabilistic models for classification, both of which make the *naive Bayes assumption*. Some use a multi-variate Bernoulli model, that is, a Bayesian Network with no dependencies between words and binary word features (*e.g.* Larkey and Croft 1996; Koller and Sahami 1997). Others use a multinomial

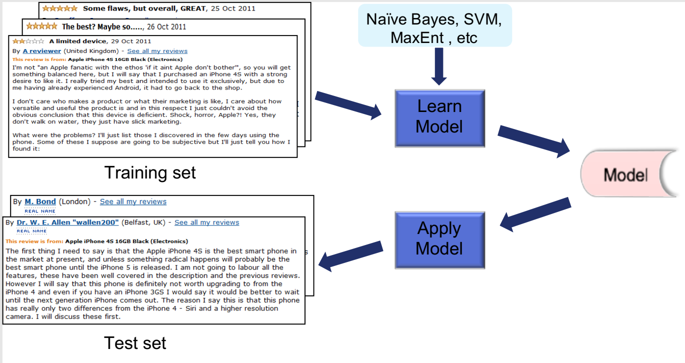
learning, especially when the number of attributes is large.

Document classification is just such a domain with a large number of attributes. The attributes of the examples to be classified are words, and the number of different words can be quite large indeed. While some simple document classification tasks can be accurately performed with vocabulary sizes less than one

# Supervised Classification

- **Supervised learning:** the machine learning task of inferring a function from labeled training data
- Given:
  - **Target:** a fixed set of **classes**: $Y = y_1, y_2, \ldots, y_n$, e.g. {sports, politics, ..., music}
  - **Training data:** a collection of data objects X with known classes Y, i.e. $(X, Y) = (x_1, y_1), (x_2, y_2) \ldots (x_n, , y_n)$. E.g {(d1, sports), (d2, sports), (d3, music) ...}.
  - **Testing data:** a description of an unseen instance, Dnew e.g. a new document **without** class label information
- Goal:
  - Predict the category/class of $D_{new} : y(x) \in Y$, where $y(x)$ is a *classification function*, aka *trained model*, whose domain is X and whose range is $Y$.

# Supervised Classifier




- Rely on syntactic or co-occurrence patterns in large text corpora

# The Bayes Rule

$$\underbrace{p(Y \mid X_1,\ldots,X_n)}_{\text{Posterior}} = \frac{\overbrace{P(X_1,\ldots,X_n \mid Y)}^{\text{Likelihood}}\,\overbrace{P(Y)}^{\text{Prior}}}{\underbrace{P(X_1,\ldots,X_n)}_{\text{Normalization Constant}}}$$

- $P(Y)$: Prior belief (probability of hypothesis $Y$ before seeing any data)
- $P(X_1, ..., X_n|Y)$: Likelihood (probability of the data if the hypothesis $Y$ is true)
- $P(X_1, ..., X_n)$: Data evidence (marginal probability of data)
- $P(Y|X_1, ..., X_n)$: Posterior (probability of hypothesis $Y$ after having seen the data)

# The Independence Assumption

- Assume A and B are Boolean Random variables. Then

"A and B are independent"

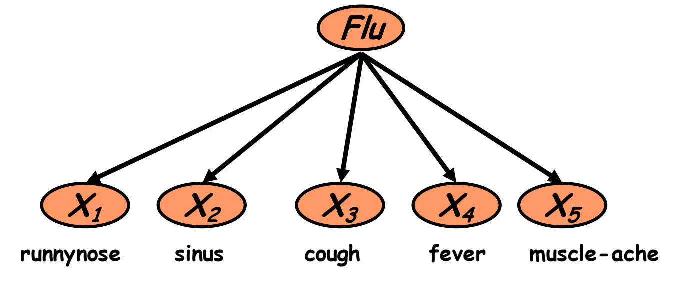
if and only if

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

"A and B are independent" is often notated as

$$A \perp B$$

# The Independence Assumption




- Features (term presence) are *independent* of each other given the class:

$$P(X_1, ..., X_5|Y) = P(X_1|Y) \bullet P(X_2|Y) \bullet ... \bullet P(X_5|Y)$$

# A corpus-based approach to SA - Machine Learning

**Naive Bayes** classifier: estimate the probability of each class given a text:

- Compute the posterior probability (Bayes rule) of each class $c_i$ for text segment $T$

$$P(c_i|T) = \frac{P(T|c_i)P(c_i)}{P(T)}$$

- Assumption of independence between features ("naive" assumption)

$$P(T|c_i) = P(t_1, t_2, ..., t_j|c_i) \approx \prod_{j=1}^{n} P(t_j|c_i)$$

where $T$ is described by a number of attributes or features $t_1, ..., t_j$

I.e. joint probability of the features given the class is approximated by the product of the probabilities of each feature given the class.

A Naive Bayes classifier (ctd)

- **Likelihood**: product of probabilities of each feature value of segment occurring with class $c_i$

$$\prod_{j=1}^{n} P(t_j|c_i)$$

- **Prior**: probability of segment having class $c_i$

$$P(c_i)$$

- **Evidence**: product of probabilities of features of segment – constant term for all classes, so can be disregarded:

$$\prod_{j=1}^{n} P(t_j)$$

**Final decision**:

$$\operatorname*{argmax}_{c_i} \prod_{j=1}^{n} P(t_j|c_i)P(c_i) \quad = \operatorname*{argmax}_{c_i} P(c_i) \prod_{j=1}^{n} P(t_j|c_i)$$

# A corpus-based approach to SA - Machine Learning

A Naive Bayes classifier - a worked out example

- Corpus of movie reviews: 7 examples for **training**

| Doc | Words | Class |
|---|---|---|
| 1 | Great movie, excellent plot, renowned actors | Positive |
| 2 | I had not seen a fantastic plot like this in good 5 years. Amazing!!! | Positive |
| 3 | Lovely plot, amazing cast, somehow I am in love with the bad guy | Positive |
| 4 | Bad movie with great cast, but very poor plot and unimaginative ending | Negative |
| 5 | I hate this film, it has nothing original | Negative |
| 6 | Great movie, but not... | Negative |
| 7 | Very bad movie, I have no words to express how I dislike it | Negative |

A Naive Bayes classifier - a worked out example (ctd)

- **Features**: adjectives (bag-of-words)

| Doc | Words | Class |
|---|---|---|
| 1 | Great movie, excellent plot, renowned actors | Positive |
| 2 | I had not seen a fantastic plot like this in good 5 years. amazing !!! | Positive |
| 3 | Lovely plot, amazing cast, somehow I am in love with the bad guy | Positive |
| 4 | Bad movie with great cast, but very poor plot and unimaginative ending | Negative |
| 5 | I hate this film, it has nothing original. Really bad | Negative |
| 6 | Great movie, but not... | Negative |
| 7 | Very bad movie, I have no words to express how I dislike it | Negative |

# A corpus-based approach to SA - Machine Learning

Relative frequency in corpus is the simplest approach to estimating probabilities:

**Priors**:

$$P(positive) = count(positive)/N = 3/7 = 0.43$$

$$P(negative) = count(negative)/N = 4/7 = 0.57$$

where N = total training examples

Assume standard pre-processing: tokenisation, lowercasing, punctuation removal (except special punctuation like !!!)

# A corpus-based approach to SA - Machine Learning

**Likelihoods**:

$$P(t_j|c_i) = \frac{count(t_j, c_i)}{count(c_i)}$$

Count word $t_j$ in class $c_i$ / total words in that class

| | | | |
|---|---|---|---|
| $P(amazing\|positive)$ | $= 2/10$ | $P(amazing\|negative)$ | $= 0/8$ |
| $P(bad\|positive)$ | $= 1/10$ | $P(bad\|negative)$ | $= 3/8$ |
| $P(excellent\|positive)$ | $= 1/10$ | $P(excellent\|negative)$ | $= 0/8$ |
| $P(fantastic\|positive)$ | $= 1/10$ | $P(fantastic\|negative)$ | $= 0/8$ |
| $P(good\|positive)$ | $= 1/10$ | $P(good\|negative)$ | $= 0/8$ |
| $P(great\|positive)$ | $= 1/10$ | $P(great\|negative)$ | $= 2/8$ |
| $P(lovely\|positive)$ | $= 1/10$ | $P(lovely\|negative)$ | $= 0/8$ |
| $P(original\|positive)$ | $= 0/10$ | $P(original\|negative)$ | $= 1/8$ |
| $P(poor\|positive)$ | $= 0/10$ | $P(poor\|negative)$ | $= 1/8$ |
| $P(renowned\|positive)$ | $= 1/10$ | $P(renowned\|negative)$ | $= 0/8$ |
| $P(unimaginative\|positive)$ | $= 0/10$ | $P(unimaginative\|negative)$ | $= 1/8$ |
| $P(!!!\|positive)$ | $= 1/10$ | $P(!!!\|negative)$ | $= 0/8$ |

# A corpus-based approach to SA - Machine Learning

- Relative frequencies for prior ($P(c_i)$) and likelihood ($P(t_j|c_i)$) make the **model** in a Naive Bayes classifier.
- At decision (test) time, given a new segment to classify, this model is applied to find the most likely class for the segment:

$$\underset{c_i}{\operatorname{argmax}} P(c_i) \prod_{j=1}^{n} P(t_j|c_i)$$

# A corpus-based approach to SA - Machine Learning

Given a new segment to classify (**test time**):

| Doc | Words | Class |
|---|---|---|
| 8 | This was a fantastic story, good, lovely | ??? |

**Final decision**

$$\underset{c_i}{\operatorname{argmax}} P(c_i) \prod_{j=1}^{n} P(t_j|c_i)$$

$$P(positive) * P(fantastic|positive) * P(good|positive) * P(lovely|positive)$$

$$3/7 * 1/10 * 1/10 * 1/10 = 0.00043$$

---

$$P(negative) * P(fantastic|negative) * P(good|negative) * P(lovely|negative)$$

$$4/7 * 0/8 * 0/8 * 0/8 = 0$$

---

**So:** *sentiment = positive*

# A corpus-based approach to SA - Machine Learning

Given a new segment to classify (**test time**):

| Doc | Words | Class |
| --- | --- | --- |
| 9 | Great plot, great cast, great everything | ??? |

**Final decision**

$$P(positive) * P(great|positive) * P(great|positive) * P(great|positive)$$

$$3/7 * 1/10 * 1/10 * 1/10 = 0.00043$$

---

$$P(negative) * P(great|negative) * P(great|negative) * P(great|negative)$$

$$4/7 * 2/8 * 2/8 * 2/8 = 0.00893$$

---

**So:** $sentiment = negative$

# A corpus-based approach to SA - Machine Learning

What if the new segment to classify (**test time**) is:

| Doc | Words | Class |
|---|---|---|
| 10 | Lovely plot, excellent cast, amazing everything | ??? |

**Final decision**

$$P(positive) * P(lovely|positive) * P(excellent|positive) * P(amazing|positive)$$

$$3/7 * 1/10 * 1/10 * 1/10 = 0.00043$$

---

$$P(negative) * P(lovely|negative) * P(excellent|negative) * P(amazing|negative)$$

$$4/7 * 0/8 * 0/8 * 0/8 = 0$$

---

**So:** $sentiment = positive$

# A corpus-based approach to SA - Machine Learning

But if the new segment to classify (**test time**) is:

| Doc | Words | Class |
|---|---|---|
| 11 | Boring movie, annoying plot, unimaginative ending | ??? |

**Final decision**

$$P(positive) * P(boring|positive) * P(annoying|positive) * P(unimaginative|positive)$$

$$3/7 * 0/10 * 0/10 * 0/10 = 0$$

---

$$P(negative) * P(boring|negative) * P(annoying|negative) * P(unimaginative|negative)$$

$$4/7 * 0/8 * 0/8 * 1/8 = 0$$

---

**So:** $sentiment = ???$

# A corpus-based approach to SA - Machine Learning

Add smoothing to feature counts (add 1 to every count). **Likelihoods** =

$$P(t_j|c_i) = \frac{count(t_j, c_i) + 1}{count(c_i) + |V|}$$

where $|V|$ is the number of distinct attributes in training (all classes) = **12**

| Doc | Words | Class |
|---|---|---|
| 12 | Boring movie, annoying plot, unimaginative ending | ??? |

**Final decision**

$$\frac{P(positive) * P(boring|positive) * P(annoying|positive) * P(unimaginative|positive)}{}$$

$$3/7 * ((0+1)/(10+12)) * ((0+1)/(10+12)) * ((0+1)/(10+12)) = 0.000040$$

$$P(negative) * P(boring|negative) * P(annoying|negative) * P(unimaginative|negative)$$

$$4/7 * ((0+1)/(8+12)) * ((0+1)/(8+12)) * ((1+1)/(8+12)) = 0.000143$$

**So:** *sentiment = negative*

# A corpus-based approach to SA - Machine Learning

Given a trained classifier that classifies arbitrary segments of text we can use it to:

- Classify entire documents, e.g an entire review.
- Classify sentences in a document (perhaps just those identified as subjective) and then compute a classification of the document by aggregating the sentiments of individual sentences, according to some function.
- Classify sentences or phrases identified as discussing an aspect/feature of a target object (e.g. a sentence discussing battery life of a phone) and interpret the sentiment as the sentiment of opinion holder towards the specific aspect under discussion

**Questions:**

- Is this a good solution? Is it robust?
- What is the role of the prior?
- How can we improve this solution?
    - ◇ Other features? Are we missing out critical information?
    - ◇ Other algorithms?
- What about non-binary classification (e.g. 5-grades of sentiment)?

# A corpus-based approach to SA - Machine Learning

**Questions:**

- Is this a good solution? Is it robust?
  - → It's simple and will work well if data is not sparse
- What is the role of the prior?
  - → Prior is very important esp. on biased cases
- How can we improve this solution?
  - ◇ Other features? Are we missing out critical information?
    - → Using all words (in Naive Bayes) works well in some tasks
    - → Finding subsets of words may help in other tasks
    - → Using only adjectives can be limiting. Verbs like hate, dislike; nouns like love; words for inversion like not; intensifiers like very
    - → Pre-built polarity lexicons can be helpful
    - → Negation is important
  - ◇ Other algorithms?
    - → MaxEnt & SVM tend to do better than Naive Bayes
- What about non-binary classification (e.g. 5-grades of sentiment)?
  - → 5-class ordinal classification or regression algorithms can be used

# Evaluation

How do we quantify how well our Sentiment Analysis systems work?

- Create experimental datasets (aka test corpora): i.e., text segments that have been classified by humans, e.g. positive vs negative
- Compare (positive vs negative) system to human classifications
- Compute metrics like

$$\text{Accuracy} = \frac{\#\ \text{correctly classified texts}}{\#\ \text{texts}}$$

$$\text{Precision Pos} = \frac{\#\ \text{texts correctly classified as positive}}{\#\ \text{texts classified as positive}}$$

$$\text{Recall Pos} = \frac{\#\ \text{texts correctly classified as positive}}{\#\ \text{positive texts}}$$

$$\text{F-measure Pos} = \frac{2 * \text{Precision Pos} * \text{Recall Pos}}{\text{Precision Pos} + \text{Recall Pos}}$$

Same for negative class.

**Baseline**: most frequent class in the training set.

# Conclusions

- Naïve Bayes classifier:
  - Really easy to implement and often works well
  - Often a good first thing to try
- Actually, the Naïve Bayes assumption is almost never true
- Still, Naïve Bayes often performs surprisingly well even when its assumption does not hold
- SA is an exciting topic, many applications, huge market for systems, particularly in focused domains.
- Promising results with simple techniques, but many interesting research challenges to be addressed for high accuracy.

# Extra reading

Bing Liu and Lei Zhang (2012). A survey on opinion mining and sentiment analysis. Kluwer Academic Publishers: `http://www.cs.uic.edu/~lzhang3/paper/opinion_survey.pdf`

Bing Liu (2012). Sentiment Analysis and Opinion Mining. Morgan and Claypool Publishers. Draft on line at: `https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf`

Article on SemEval in Wikipedia: `https://en.wikipedia.org/wiki/SemEval`.