ANLP Lecture 7
Text Categorization with Naive Bayes

Sharon Goldwater

30 September 2019

School of informatics

Text categorization: example

Dear Prof.Sharon Goldwater:

My name is [XX]. I am an ambitious applicant for the Ph.D program of Electrical Engineering and Computer Science at your university. Especially being greatly attracted by your research projects and admiring for your achievements via the school website, I cannot wait to write a letter to express my aspiration to undertake the Ph.D program under your supervision.

I have completed the M.S. program in Information and Communication Engineering with a high GPA of 3.95/4.0 at [YY] University. In addition to throwing myself into the specialized courses in [...] I took part in the research projects, such as [...]. I really enjoyed taking the challenges in the process of the researches and tests, and I spent two years on the research project [...]. We proved the effectiveness of the new method for [...] and published the result in [...].

Having read your biography, I found my academic background and research experiences indicated some possibility of my qualification to join your team. It is my conviction that the enlightening instruction, cutting-edge research projects and state of-the-art facilities offered by your team will direct me to make breakthroughs in my career development in the arena of electrical engineering and computer science. Thus, I shall be deeply grateful if you could give me the opportunity to become your student. Please do not hesitate to contact me, should you need any further information about my scholastic and research experiences.

Yours sincerely, [XX].

Today's lecture

- What are some examples of text categorization tasks?
- What is a Naive Bayes classifier and how do we apply it to text categorization (in general, or for specific tasks)?
- What are some pros and cons of Naive Bayes?
- How do we evaluate categorization accuracy?

Text categorization (classification)

We might want to categorize the *content* of the text:

- Spam detection (binary classification: spam/not spam)
- Sentiment analysis (binary or multiway)
  - movie, restaurant, product reviews (pos/neg, or 1-5 stars)
  - political argument (pro/con, or pro/con/neutral)
- Topic classification (multiway: sport/finance/travel/etc)

Text categorization (classification)

Or we might want to categorize the *author* of the text (**authorship attribution**):

- Native language identification (e.g., to tailor language tutoring)
- Diagnosis of disease (psychiatric or cognitive impairments)
- Identification of gender, dialect, educational background (e.g., in forensics [legal matters], advertising/marketing).

Formalizing the task

- Given document d and set of categories C, we want to assign d to the most probable category $\hat{c}$:

$$\begin{aligned}\hat{c} &= \operatorname*{argmax}_{c \in C} P(c|d) \\ &= \operatorname*{argmax}_{c \in C} \frac{P(d|c)P(c)}{P(d)} \\ &= \operatorname*{argmax}_{c \in C} P(d|c)P(c)\end{aligned}$$

Document model

Each document d is represented by **features** $f_1, f_2, \dots f_n$, e.g.:

- For topic classification: 2000 most frequent words, excluding **stopwords** like *the, a, do, in*.
- For sentiment classification: words from a **sentiment lexicon**

In fact, we only care about the feature *counts*, so this is a **bag-of-words** (unigram) model.

Task-specific features

Example words from a **sentiment lexicon**:

Positive:

absolutely
adorable
accepted
acclaimed
accomplish
achieve
action
active
admire
adventure
affirm
...

beaming
beautiful
believe
beneficial
bliss
bountiful
bounty
brave
bravo
brilliant
bubbly

calm
celebrated
certain
champ
champion
charming
cheery
choice
classic
classical
clean
...

Negative:

abysmal
adverse
alarming
angry
annoy
anxious
apathy
appalling
atrocious
awful

bad
banal
barbed
belligerent
bemoan
beneath
boring
broken

callous
can't
clumsy
coarse
cold
collapse
confused
contradictory
contrary
corrosive
corrupt
...

From `http://www.enchantedlearning.com/wordlist/`

Example documents

- Possible feature counts from training documents in a spam-detection task (where we did not exclude stopwords):

	the	your	model	cash	Viagra	class	account	orderz
doc 1	12	3	1	0	0	2	0	0
doc 2	10	4	0	4	0	0	2	0
doc 3	25	4	0	0	0	1	1	0
doc 4	14	2	0	1	3	0	1	1
doc 5	17	5	0	2	0	0	1	1

Document model, cont.

- Representing d using its features gives us:

$$P(d|c) = P(f_1, f_2, \dots f_n|c)$$

- But we can't estimate this joint probability well (too sparse).
- So, make a **Naive Bayes** assumption: features are conditionally independent given class.

$$P(d|c) \approx P(f_1|c)P(f_2|c) \dots P(f_n|c)$$

Full model

- Given document with features $f_1, f_2, \dots f_n$ and set of categories C, choose

$$\hat{c} \;=\; \operatorname*{argmax}_{c \in C} P(c) \prod_{i=1}^{n} P(f_i|c)$$

- This is called a **Naive Bayes classifier**
  - see Basic Prob Theory reading Ex 5.5.3 for a non-text example

Generative process

- Naive Bayes classifier is another generative model.
- Assumes the data (features in each doc) were generated as
  - For each document, choose its class c with prob $P(c)$.
  - For each feature in each doc, choose the value of that feature with prob $P(f|c)$

Learning the class priors

- $P(c)$ normally estimated with MLE:

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

  - N_c = the number of training documents in class c
  - N = the total number of training documents

Learning the class priors: example

- Given training documents with correct labels:

	the	your	model	cash	Viagra	class	account	orderz	spam?
doc 1	12	3	1	0	0	2	0	0	-
doc 2	10	4	0	4	0	0	2	0	+
doc 3	25	4	0	0	0	1	1	0	-
doc 4	14	2	0	1	3	0	1	1	+
doc 5	17	5	0	2	0	0	1	1	+

- $\hat{P}(\text{spam}) = 3/5$

Learning the feature probabilities

- $P(f_i|c)$ normally estimated with simple smoothing:

$$\hat{P}(f_i|c) = \frac{\text{count}(f_i, c) + \alpha}{\sum_{f \in F}(\text{count}(f, c) + \alpha)}$$

  - $\text{count}(f_i, c)$ = the number of times f_i occurs in class c
  - F = the set of possible features
  - α: the smoothing parameter, optimized on held-out data

Learning the feature probabilities: example

	the	your	model	cash	Viagra	class	account	orderz	spam?
doc 1	12	3	1	0	0	2	0	0	-
doc 2	10	4	0	4	0	0	2	0	+
doc 3	25	4	0	0	0	1	1	0	-
doc 4	14	2	0	1	3	0	1	1	+
doc 5	17	5	0	2	0	0	1	1	+

Learning the feature probabilities: example

	the	your	model	cash	Viagra	class	account	orderz	spam?
doc 1	12	3	1	0	0	2	0	0	-
doc 2	10	**4**	0	4	0	0	2	0	+
doc 3	25	4	0	0	0	1	1	0	-
doc 4	14	**2**	0	1	3	0	1	1	+
doc 5	17	**5**	0	2	0	0	1	1	+

$$\hat{P}(\text{your}|+) = \frac{(4+2+5+\alpha)}{(\text{tokens in + class})+\alpha|F|} = (11+\alpha)/(68+\alpha|F|)$$

Learning the feature probabilities: example

	the	your	model	cash	Viagra	class	account	orderz	spam?
doc 1	12	3	1	0	0	2	0	0	-
doc 2	10	4	0	4	0	0	2	0	+
doc 3	25	4	0	0	0	1	1	0	-
doc 4	14	2	0	1	3	0	1	1	+
doc 5	17	5	0	2	0	0	1	1	+

$$\hat{P}(\text{your}|+) = \frac{(4+2+5+\alpha)}{(\text{tokens in + class})+\alpha|F|} = (11+\alpha)/(68+\alpha|F|)$$

$$\hat{P}(\text{your}|-) = \frac{(3+4+\alpha)}{(\text{tokens in } - \text{ class})+\alpha|F|} = (7+\alpha)/(49+\alpha|F|)$$

$$\hat{P}(\text{orderz}|+) = \frac{(2+\alpha)}{(\text{tokens in + class})+\alpha|F|} = (2+\alpha)/(68+\alpha|F|)$$

Classifying a test document: example

- Test document d:

  get your cash and your orderz

- Suppose there are no other features besides those in previous table (so get and and are not counted). Then

$$\begin{aligned} P(+|d) &\propto P(+)\prod_{i=1}^{n} P(f_i|+) \\ &= \frac{3}{5}\cdot\frac{11+\alpha}{(68+\alpha F)}\cdot\frac{7+\alpha}{(68+\alpha F)} \\ &\quad \cdot\frac{11+\alpha}{(68+\alpha F)}\cdot\frac{2+\alpha}{(68+\alpha F)} \end{aligned}$$

Classifying a test document: example

- Test document d:

  get your cash and your orderz

- Do the same for $P(-|d)$

- Choose the one with the larger value

Alternative feature values and feature sets

- Use only **binary** values for f_i: did this word occur in d or not?
- Use only a subset of the vocabulary for F
  - Ignore **stopwords** (function words and others with little content)
  - Choose a small task-relevant set (e.g., using a sentiment lexicon)
- Use more complex features (bigrams, syntactic features, morphological features, ...)

Task-specific features

Example words from a **sentiment lexicon**:

Positive:

absolutely
adorable
accepted
acclaimed
accomplish
achieve
action
active
admire
adventure
affirm
. . .

beaming
beautiful
believe
beneficial
bliss
bountiful
bounty
brave
bravo
brilliant
bubbly

calm
celebrated
certain
champ
champion
charming
cheery
choice
classic
classical
clean
. . .

Negative:

abysmal
adverse
alarming
angry
annoy
anxious
apathy
appalling
atrocious
awful

bad
banal
barbed
belligerent
bemoan
beneath
boring
broken

callous
can't
clumsy
coarse
cold
collapse
confused
contradictory
contrary
corrosive
corrupt
. . .

From `http://www.enchantedlearning.com/wordlist/`

Task-specific features

- But: other words might be relevant for specific sentiment analysis tasks.
  - E.g., quiet, memory for product reviews.
- And for other tasks, stopwords might be very useful features
  - E.g., People with schizophrenia use more 2nd-person pronouns (Watson et al., 2012), those with depression use more 1st-person (Rude et al., 2004).
- Probably better to use too many irrelevant features than not enough relevant ones.

Advantages of Naive Bayes

- Very easy to implement
- Very fast to train and test
- Doesn't require as much training data as some other methods
- Usually works reasonably well

Use as a simple baseline for any classification task.

Problems with Naive Bayes

- Naive Bayes assumption is naive!
- Consider categories TRAVEL, FINANCE, SPORT.
- Are the following features independent given the category?

  beach, sun, ski, snow, pitch, palm, football, relax, ocean

Problems with Naive Bayes

- Naive Bayes assumption is naive!
- Consider categories TRAVEL, FINANCE, SPORT.
- Are the following features independent given the category?

  beach, sun, ski, snow, pitch, palm, football, relax, ocean

- No! They might be closer if we defined finer-grained categories (beach vacations vs. ski vacations), but we don't usually want to.

Non-independent features

- Features are not usually independent given the class
- Adding multiple feature types (e.g., words and morphemes) often leads to even stronger correlations between features
- Accuracy of classifier can sometimes still be ok, but it will be highly **overconfident** in its decisions.
  - Ex: NB sees 5 features that all point to class 1, treats them as five independent sources of evidence.
  - Like asking 5 friends for an opinion when some got theirs from each other.

How to evaluate performance?

- Important question for any NLP task
- **Intrinsic** evaluation: design a measure inherent to the task
  - Language modeling: perplexity
  - POS tagging: accuracy (% of tags correct)
  - Categorization: F-score (coming up next)

How to evaluate performance?

- Important question for any NLP task
- **Intrinsic** evaluation: design a measure inherent to the task
  - Language modeling: perplexity
  - POS tagging: accuracy (% of tags correct)
  - Categorization: F-score (coming up next)
- **Extrinsic** evaluation: measure effects on a downstream task
  - Language modeling: does it improve my ASR/MT system?
  - POS tagging: does improve my parser/IR system?
  - Categorization: does it reduce user search time in an IR setting?

Intrinsic evaluation for categorization

- Categorization as detection: document about sport or not?
- Classes may be very unbalanced.
- Can get 95% accuracy by always choosing "not"; but this isn't useful.
- Need a better measure.

Two measures

- Assume we have a **gold standard**: correct labels for test set
- We also have a system for detecting the items of interest (docs about sport)

$$\text{Precision} = \frac{\#\text{ items detected and was right}}{\#\text{ items system detected}}$$

$$\text{Recall} = \frac{\#\text{ items detected and was right}}{\#\text{ items system should have detected}}$$

Example of precision and recall

	Doc about sports?										
Gold standard	**Y**	**Y**	N	N	**Y**	N	N	N	**Y**	N	N
System output	N	**Y**	N	**Y**	N	N	N	N	**Y**	N	N

- # 'Y' we got right = 2
- # 'Y' we guessed = 3
- # 'Y' in GS = 4

- Precision = 2/3
- Recall = 2/4

Why use both measures?

Systems often have (implicit or explicit) tuning thresholds on how many answers to return.

- e.g., Return as **Y** all docs where system thinks P(C=sport) is greater than t.
- Raise t: higher precision, lower recall.
- Lower t: lower precision, higher recall.

Doc	Sys prob	Gold
23	0.99	**Y**
12	0.98	**Y**
45	0.93	**Y**
01	0.93	**Y**
37	0.89	N
24	0.84	**Y**
16	0.78	**Y**
18	0.75	N
20	0.72	**Y**
…	…	…
38	0.03	N
19	0.03	N

Precision-Recall curves

- If system has a tunable parameter to vary the precision/recall:

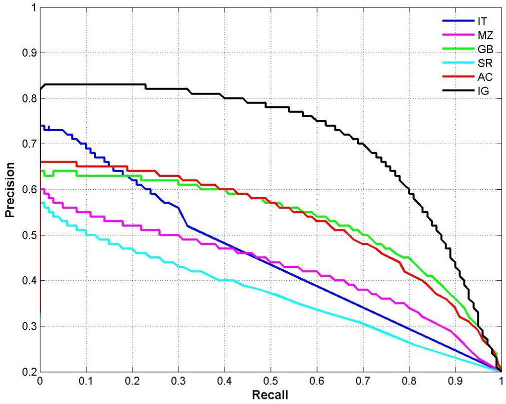

Figure from: http://ivrgwww.epfl.ch/supplementary_material/RK_CVPR09/

F-measure

- Can also combine precision and recall into single **F-measure**:

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

- Normally we just set $\beta = 1$ to get F_1:

$$F_1 = \frac{2PR}{P+R}$$

- F_1 is the harmonic mean of P and R: similar to arithmetic mean when P and R are close, but penalizes large differences between P and R.

Questions for review

- What are some examples of text categorization tasks?
- What is a Naive Bayes classifier and how do we apply it to text categorization (in general, or for specific tasks)?
- What are some pros and cons of Naive Bayes?
- How do we evaluate categorization accuracy?

Questions and exercises

1. Do the exercises at the end of Chapter 4 in JM3.
2. Why is it often ok to use MLE for estimating class probabilities? In what situation(s) might it be advisable to use smoothing?
3. Why is it often ok to use simple (add-alpha) smoothing for these models, when it's so bad for language modelling? (Hint: consider what feature sets are often used: see slides 6-7, 20-22)
4. Look at the example system output on slide 32, which shows the input documents ranked by the system's posterior probability of the document being in the 'sport' topic. Suppose there are 3 additional documents about sport in the part of the list that isn't shown. If we set the threshold $t = 0.8$, what are the precision and recall scores of this system? What about with $t = 0.9$? What is the F-score in each case?

References

Rude, S., Gortner, E.-M., and Pennebaker, J. (2004). Language use of depressed and depression-vulnerable college students. *Cognition & Emotion*, 18(8):1121–1133.

Watson, A. R., Defterali, Ç., Bak, T. H., Sorace, A., McIntosh, A. M., Owens, D. G., Johnstone, E. C., and Lawrie, S. M. (2012). Use of second-person pronouns and schizophrenia. *The British Journal of Psychiatry*, 200(4):342–343.