Lecture 7
Naive Bayes and Sentiment Classification

CS 6320

Naive Bayes and Sentiment Classification

- Classification is a basic activity of human life.
  Recognize and classify. Examples: classify people, images, voices, assigning grades, emails, etc.

- Text classification/categorization – is the task of assigning a label (category) to a document ; it is a basic NLP capability.

- Sentiment analysis- is an example of text categorization – it assigns a positive or negative label to text.

  Examples: Product reviews, movie review, book review, restaurant reviews, political commentaries, etc.

Other text classification problems

- Spam detection
- Language identification

Naive Bayes and Sentiment Classification

- Example
  - + *...richly applied satire, and some great plot twists*
  - _ *It was pathetic. The worst part about it was the boxing scenes...*
  - + *...awesome caramel sauce and sweet toasty almonds. I love this place!*
  - _ *...awful pizza and ridiculously overpriced...*

Approach to classification: supervised machine learning

- Take an input *x* and a fixed set of output classes $Y = y , y_2 ,\dots, y_M$ and return a predicted class $y \in Y$.

For texts: c is a class, (instead of *y*) and d – document.

Training examples $(d_1 , c_1), (d_2 , c_2) \dots .(d_N , C_N)$

Goal: classify a new document d to its correct class $c \in C$.

Probabilistic classifiers

1. Generation classifiers – given an observation return the class most likely to have generated the observation.

   Example: Naive Bayes

2. Discriminative classifiers – learn what features from input are most useful to discriminate between possible classes.

   Example: logistic regression

Naive Bayes Classifiers

- Given a document d and set of classes $c \in C$, returns

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

$$= \underset{c \in C}{\operatorname{argmax}} \frac{P(d|c)\, P(c)}{P(d)}$$

$$= \underset{c \in C}{\operatorname{argmax}} \quad \underset{\substack{\uparrow \\ \text{likelihood} \\ \text{probability}}}{P(d|c)} \ \underset{\substack{\uparrow \\ \text{prior} \\ \text{probability}}}{P(c)}$$

Naive Bayes Classifiers

- A document *d* is represented by a set of features $f_1, f_2, \ldots, f_n$

$$\hat{c} = argmax\, P\,(f_1, f_2, \ldots, f_n | c)\, P(c)$$

- Naive Bayes assumption: Features are independent,
  ie: $P(f_i|c)$ are independent.

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

$$c_{NB} = \operatorname*{argmax}_{c \in C} P(c) \prod_{f \in F} P(f|c)$$

Naive Bayes Classifiers

- Bag -of-words approach: Features use words.

- Words are specified by their positions in the document;
  w_i– word at position i

$$C_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{i \in positions} P(w_i|c)$$

- Use log space to avoid underflow and increase speed.

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} \left[logP(c) + \textstyle\sum_{ie\ positions} \log P(w_i|c)\right]$$

Training the Naïve Bayes Classifiers

- How to learn $P(c)$ and $P(f_i|c)$?

  N_c – percentage of documents in training set with class c.

  N_{doc} - total number of documents.

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

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

  V - vocabulary for all classes.

  Example: $\hat{P}(fantastic|+) = \frac{count(fantastic,+)}{\sum_{w \in V} count(w,+)}$

- Use Laplace smoothing

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

Example

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	S=predictable with no fun

$P(-) = \frac{3}{5}$ $\qquad$ $P(+) = \frac{2}{5}$

$P(\text{"}predictable\text{"}|-) = \frac{1+1}{14+20}$ $\qquad$ $P(\text{"}predictable\text{"}|+) = \frac{0+1}{9+20}$

$P(\text{"no"}|-) = \frac{1+1}{14+20}$ $\qquad$ $P(\text{"no"}|+) = \frac{0+1}{9+20}$

$P(\text{"fun"}|-) = \frac{0+1}{14+20}$ $\qquad$ $P(\text{"fun"}|+) = \frac{1+1}{9+20}$

$$P(-)\mathrm{P}(\mathrm{S}|-) = \frac{3}{5}\mathrm{x}\frac{2 \text{ x } 2 \text{ x } 1}{34^3} = 6.1 \text{ x } 10^{-5}$$

$$P(+)P(S|+) = \frac{2}{5}\mathrm{x}\frac{1 \text{ x } 1 \text{ x } 2}{29^3} = 3.2 \text{ x } 10^{-5}$$

→ Pick Negative class

Improvements

- Fix negations

  didn't like this movie

  didn't becomes *Not_like,*

  Not_this,

  Not_movie w

- Sentiment lexicons – list of words that are pre-annotated with positive and negative sentiment.
  Examples: MPQA subjectivity lexicon; 6885 words, 2718 positive and 4912 negative; with strong or weak bias.

  \+ *admirable, beautiful, confident, dazzling, great*…

  \- *awful, bad, cheat, deny, envious, foul, hate*…

Naive Bayes as a Language Model

- A language model predicts the next word after seeing a string of words; and calculates probability of strings.
- Naïve Bayes assigns a probability to each word P(word|c). It can also assign a probability to each sentence

$$P(s \mid c) = \prod_{i \in positions} P(w_i \mid c)$$

	P(w|+)	P(w|-)
I	0.1	0.2
love	0.1	0.001
this	0.01	0.01
fun	0.05	0.005
film	0.1	0.1

$$P(\text{"}I\ love\ this\ fun\ film\text{"}|+) = 0.1x0.1x0.01x0.05x0.1 = 0.0000005$$

$$P(\text{"}I\ love\ this\ fun\ film\text{"}|-) = 0.2x0.001x0.01x0.005x0.1 = 0.000000001$$

$$P(s|+) > P(s|-)$$

Evaluation: Precision and Recall

gold standard labels

		gold positive	gold negative	
system output labels	system positive	true positive	false positive	$precision = \frac{tp}{tp + fp}$
	system negative	false negative	true negative	
		$recall = \frac{tp}{tp + fn}$		$accuracy = \frac{tp + tn}{tp + fp + tn + fn}$

$$Precision = \frac{true\ positives}{true\ positives + false\ positives}$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

F – measure

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

$$\beta > 1\ favors\ recall$$
$$\beta < 1\ favors\ precision$$

More than two classes

- Two Possibilities:

  1. Multi-label classification (*any-of*)

     - Each document can be assigned more than one label
     - Build separate binary classifiers for each class c, train on positive examples labeled c and negative examples not labeled c.
     - For a test document d, each classifier makes its decision independently, and multiple labels can be assigned to d.

More than two classes

2. Multinomial classification (*one-of*)

- Classes are mutually exclusive
- Build separate binary classifiers
- For a test document, run all classifiers and choose the label from the classifier with the highest score.

Example: 3-way one-of email classification

gold labels

System output		Urgent	Normal	Spam	
	urgent	8	10	1	$precision_u = \frac{8}{8+10+1}$
	normal	5	60	50	$precision_n = \frac{60}{5+60+50}$
	spam	3	30	200	$precision_s = \frac{200}{3+30+200}$
		recallu= $\frac{8}{8+5+3}$	recalln= $\frac{60}{10+60+30}$	recalls= $\frac{200}{1+50+200}$	

Example: 3-way one-of email classification

Class 1: Urgent

	true urgent	true not
System urgent	8	11
System not	8	340

$$precision = \frac{8}{8+11} = .42$$

Class 2: Normal

	true normal	true not
System normal	60	55
System not	40	212

$$precision = \frac{60}{60+55} = .52$$

Class 3: Spam

	true spam	true not
System spam	200	33
System not	51	83

$$precision = \frac{200}{200+33} = .86$$

Pooled

	true yes	true no
System yes	268	99
System no	99	635

$$\begin{matrix} microaverage \\ precision \end{matrix} = \frac{268}{268+99} = .73$$

$$\begin{matrix} macroaverage \\ precision \end{matrix} = \frac{.42+.52+.86}{3} = .60$$