

Natural Language Processing (NLP)

Today's objectives and plan

Morning: NLP and Information Retrieval

- Text Featurization
 - Text processing pipeline
 - Bag of words and TF-IDF
- Morning assignment: Build a basic text processing pipeline to compare NYT articles using nltk and sklearn

Afternoon: Document Classification with Naïve Bayes

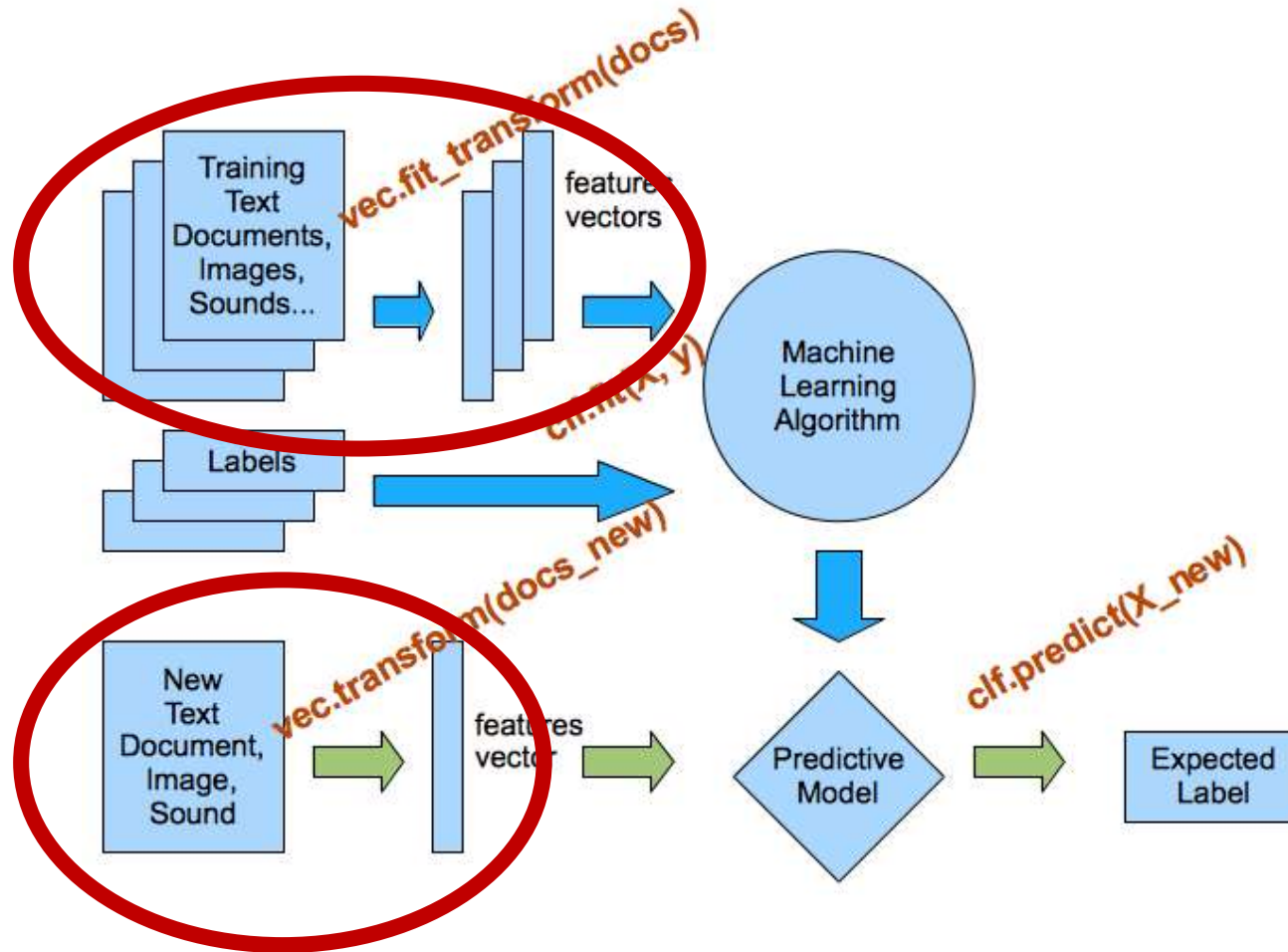
- Reviews
 - Bayes' Rule
 - MAP Estimation
 - Independence and Conditional Independence
- Naïve Bayes Classifier
- Document Classification with Naïve Bayes
- Pair programming: (1) Implementing Naive Bayes Classifier and Document Classification; (2) Applications of tf-idf and cosine similarity

Morning: NLP and Information Retrieval

Motivation

- You're Google News, and you want to group news articles by topic
- You're a legal tech firm, and you need to sift through 1,000,000 pages of legal documents to find the relevant ones

Text Featurization



Text Featurization Pipeline

- Tokenization
 - Take the document and split it into a list of tokens
- Lower case conversion
- Stop words removal
- Stemming/Lemmatization
 - Reduce words to their root form
 - Stemming (crude character-based method)
 - E.g., “walked” → “walk”
 - Lemmatization (based on the lexicon; performs better than stemming)
 - E.g., “people” → “person”, “better” → “good”
- Bag of words/N-grams

Terminology

- Corpus:
 - a dataset of text; e.g., newspaper articles, tweets
- Document:
 - a single entry from our corpus; e.g., article, sentence, tweet
- Vocabulary:
 - all the words that appear in our corpus
- Token:
 - an entity; e.g., a word

Sample sentence

Students are learning from other students

Tokenization

- Take the document and split it into a list of tokens

Students are learning from other students



['Students', 'are', 'learning', 'from', 'other', 'students']

Lower case conversion

['Students', 'are', 'learning', 'from', 'other', 'students']



['students', 'are', 'learning', 'from', 'other', 'students']

Stop word removal

- Words we ignore in our analysis that are too common to be useful (sklearn and nltk have standard list of stop words)

['students', 'are', 'learning', 'from', 'other', 'students']



['students', 'learning', 'other', 'students']

Stemming/Lemmatization

- Reduce words to their root form
 - Stemming (crude character-based method)
 - Lemmatization (based on the lexicon; performs better than stemming)

['students', 'learning', 'other', 'students']



['student', 'learn', 'other', 'student']

With a corpus made of 3 documents:

1. Students are learning from other students

1. ['student', 'learn', 'other', 'student']

2. I am teaching at Galvanize

2. ['teach', 'galvanize']

3. There are students learning at Galvanize

3. ['student', 'learn', 'galvanize']

Bag of words

- A document represented as a vector of word counts is called a “bag of words”
- Vector for our corpus:
(galvanize, learn, other, student, teach)

document	galvanize	learn	other	student	teach
['student', 'learn', 'other', 'student']					
['teach', 'galvanize']					
['student', 'learn', 'galvanize']					

Bag of words

- A document represented as a vector of word counts is called a “bag of words”
- Vector for our corpus:
(galvanize, learn, other, student, teach)

document	galvanize	learn	other	student	teach
['student', 'learn', 'other', 'student']	0	1	1	2	0
['teach', 'galvanize']	1	0	0	0	1
['student', 'learn', 'galvanize']	1	1	0	1	0

Issues with bags of words

- Bags of words are naïve
 - Word counts
 - Counts emphasize results from longer documents
 - Every word has equal weighting
 - But “other” and “student” have different predictive power
- Is there a better way to featurize?

Term frequencies (TF)

- Normalize counts within a document to frequency

$$\text{tf}(t, d) = \frac{\text{total count of term } t \text{ in document } d}{\text{total count of all terms in document } d}$$

document	galvanize	learn	other	student	teach
['student', 'learn', 'other', 'student']					
['teach', 'galvanize']					
['student', 'learn', 'galvanize']					

Term frequencies (TF)

- Normalize counts within a document to frequency

$$\text{tf}(t, d) = \frac{\text{total count of term } t \text{ in document } d}{\text{total count of all terms in document } d}$$

document	galvanize	learn	other	student	teach
['student', 'learn', 'other', 'student']	0	$\frac{1}{4} = .25$	$\frac{1}{4} = .25$	$\frac{2}{4} = .5$	0
['teach', 'galvanize']	$\frac{1}{2} = .5$	0	0	0	$\frac{1}{2} = .5$
['student', 'learn', 'galvanize']	$\frac{1}{3} = .333$	$\frac{1}{3} = .333$	0	$\frac{1}{3} = .333$	0

Issues with term frequencies

- Words found in only one document should have highest weighting
- Words found in every documents should have lowest weighting

Inverse document frequencies (IDF)

$$\text{idf}(t, D) = \log \frac{\text{total count of documents in corpus } D}{\text{count of documents containing term } t}$$

document	galvanize	learn	other	student	teach
['student', 'learn', 'other', 'student']					
['teach', 'galvanize']					
['student', 'learn', 'galvanize']					
$\text{idf}(t, D)$					

Inverse document frequencies (IDF)

$$\text{idf}(t, D) = \log \frac{\text{total count of documents in corpus } D}{\text{count of documents containing term } t}$$

document	galvanize	learn	other	student	teach
['student', 'learn', 'other', 'student']		X	X	X	
['teach', 'galvanize']	X				X
['student', 'learn', 'galvanize']	X	X		X	
$\text{idf}(t, D)$	$\log \frac{3}{2} = .405$	$\log \frac{3}{2} = .405$	$\log \frac{3}{1} = 1.10$	$\log \frac{3}{2} = .405$	$\log \frac{3}{1} = 1.10$

TF-IDF

$$\text{tf-idf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D)$$

document	galvanize	learn	other	student	teach
['student', 'learn', 'other', 'student']					
['teach', 'galvanize']					
['student', 'learn', 'galvanize']					

TF-IDF

$$\text{tf-idf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D)$$

document	galvanize	learn	other	student	teach
['student', 'learn', 'other', 'student']	0	$.25 \times .405$ $= .101$	$.25 \times 1.10$ $= .275$	$.5 \times .405$ $= .203$	0
['teach', 'galvanize']	$.5 \times .405$ $= .203$	0	0	0	$.25 \times 1.10$ $= .275$
['student', 'learn', 'galvanize']	$.333 \times .405$ $= .135$	$.333 \times .405$ $= .135$	0	$.333 \times .405$ $= .135$	0

Comparing TF-IDF vectors of documents: Cosine Similarity

$$\text{similarity} = \cos \theta = \frac{A \cdot B}{\|A\| \|B\|}$$

- ['student', 'learn', 'other', 'student'] vs. ['teach', 'galvanize']
→ (0, .101, .275, .203, 0) vs. (.203, 0, 0, 0, .275)
→ $\text{similarity} = \frac{0}{.36 \times .34} = 0$
- ['student', 'learn', 'other', 'student'] vs. ['student', 'learn', 'galvanize']
→ (0, .101, .275, .203, 0) vs. (.135, .135, 0, .135, 0)
→ $\text{similarity} = \frac{0.041}{.36 \times .23} = .34$

N-grams

- Bag of words lose word order
- What to do when it matters?
- N-grams attempt to retain some of it:
['student', 'learn', 'other', 'student']
 - (student, learn), (learn, other), (other, student) (*2-grams*)
 - (student, learn, other), (learn, other, student) (*3-grams*)

Advanced NLP Problem Types

- Sentiment analysis
 - <http://nlp.stanford.edu/sentiment/index.html>
- Machine translation
 - <https://medium.com/s-c-a-l-e/how-baidu-mastered-mandarin-with-deep-learning-and-lots-of-data-1d94032564a5>

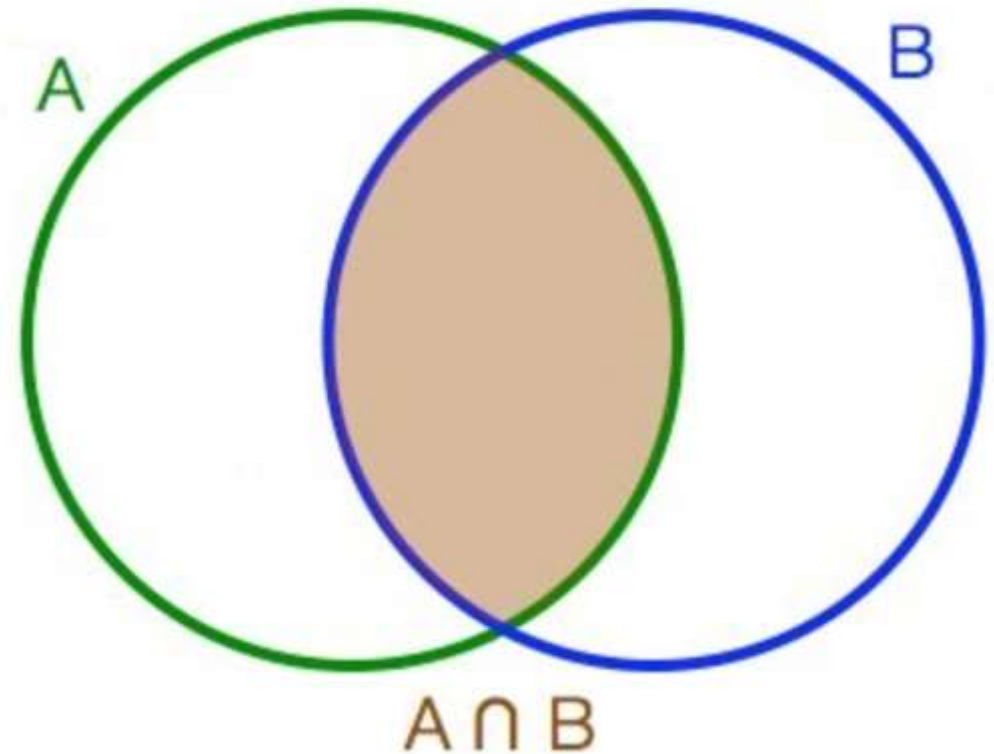
Morning assignment

Afternoon: Document Classification with Naïve Bayes

Bayes' Rule Review

Conditional Probability Review

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)}$$



Bayes' Rule

posterior
probability of
“H” given the
evidence

likelihood of
the evidence
“E” if the
hypothesis “H”
is true

prior probability
of “H”

$$P(H | E) = \frac{P(E | H)P(H)}{P(E)}$$

prior probability that the
evidence itself is true
(but also a normalizing
constant)

Bayes' Rule after expanding $P(E)$

$$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E \mid H)P(H) + P(E \mid H^c)P(H^c)}$$

- $E = E \cap (H \cup H^c) = (E \cap H) \cup (E \cap H^c)$
- $P(E) = P(E \cap H) + P(E \cap H^c)$ (independent events)
- $P(E) = P(E \mid H)P(H) + P(E \mid H^c)P(H^c)$

Poll: Relating Prior Knowledge/Belief to Data

You have a drawer of 100 coins, 10 of which are biased

$$P(\text{head} \mid \text{fair}) = .5$$

$$P(\text{head} \mid \text{biased}) = .25$$

You randomly choose a coin and flip it three times. It comes up heads all three times

- **What is $P(\text{fair} \mid \text{H, H, H})$?**

Poll: Relating Prior Knowledge/Belief to Data (cont.)

$$P(\text{fair} \mid H, H, H)$$

$$= \frac{P(H, H, H \mid \text{fair})P(\text{fair})}{P(H, H, H \mid \text{fair})P(\text{fair}) + P(H, H, H \mid \text{biased})P(\text{biased})}$$

$$= \frac{.5^3 \times .9}{.5^3 \times .9 + .25^3 \times .1} = .986$$

Relating Prior Knowledge/Belief to Data (cont.)

$$P(\text{biased} \mid H, H, H)$$

$$= \frac{P(H, H, H \mid \text{biased})P(\text{biased})}{P(H, H, H \mid \text{biased})P(\text{biased}) + P(H, H, H \mid \text{fair})P(\text{fair})}$$

$$= \frac{.25^3 \times .1}{.25^3 \times .1 + .5^3 \times .9} = .014$$

MAP Estimation Review

MAP Estimation Review

- Recall Bayes' Rule:

$$P(H | E) = \frac{P(E | H)P(H)}{P(E)}$$

- MAP finds H to maximize $P(H | E)$

$$\operatorname{argmax}_H P(H | E) = \operatorname{argmax}_H \frac{P(E | H)P(H)}{P(E)} = \operatorname{argmax}_H P(E | H)P(H)$$

(prior is constant)

Poll: Relating Prior Knowledge/Belief to Data

You have a drawer of 100 coins, 10 of which are biased

$$P(\text{head} \mid \text{fair}) = .5$$

$$P(\text{head} \mid \text{biased}) = .25$$

You randomly choose a coin and flip it three times. It comes up heads all three times

- **Which coin type (fair or unfair) is most probable under the posterior?**

Poll: Relating Prior Knowledge/Belief to Data (cont.)

$$\operatorname{argmax}_{\text{coin}} P(\text{coin} \mid H, H, H)?$$

$$\operatorname{argmax}_{\text{coin}} P(\text{coin} \mid H, H, H) = \operatorname{argmax}_{\text{coin}} P(H, H, H \mid \text{coin})P(\text{coin})$$

$$P(H, H, H \mid \text{fair})P(\text{fair}) = .5^3 \times .9 = .113$$

$$P(H, H, H \mid \text{biased})P(\text{biased}) = .25^3 \times .1 = .00156$$



$$\operatorname{argmax}_{\text{coin}} P(\text{coin} \mid H, H, H) = \text{fair}$$

Independence and Conditional Independence Review

Independence

If A and B are independent,

$$P(B \mid A) = P(B)$$

Then using Bayes' rule:

$$\frac{P(A \cap B)}{P(A)} = P(B)$$



$$P(A \cap B) = P(A)P(B)$$

Conditional Independence

$$P(A \cap B \mid C) = P(A \mid C)P(B \mid C)$$

Naïve Bayes Classifier

Derivation

$$\operatorname{argmax}_y P(y | X)?$$

$$P(y | X) = \frac{P(X | y)P(y)}{P(X)} \text{ (Bayes' rule)}$$

$$\operatorname{argmax}_y P(y | X) = \operatorname{argmax}_y P(X | y)P(y) \text{ (MAP estimation)}$$

$$P(X | y) = P(x_1 | y)P(x_2 | y) \dots P(x_p | y) = \prod_{j=1}^p P(x_j | y)$$

Naïve Bayes Classifier is MAP estimation combined with conditional independence

$$\operatorname{argmax}_y P(y \mid X) = \operatorname{argmax}_y P(y) \prod_{j=1}^p P(x_j \mid y)$$

Document Classification with Naïve Bayes

Motivation

How to predict what topic a given document is about?

E.g.,

“the cat jumped over the tree”

- Q: How can we decide whether this document is fiction or non-fiction?
- A: Use word counts from corpus of label fiction and non-fiction documents to train a Naïve Bayes classifier

Document Classification with Naïve Bayes

$$\operatorname{argmax}_y P(y \mid X) = \operatorname{argmax}_y P(y) \prod_{j=1}^p P(x_j \mid y)$$

↓

$$\operatorname{argmax}_{\text{topic}} P(\text{topic} \mid \text{document}) = \operatorname{argmax}_{\text{topic}} P(\text{topic}) \prod_{\text{word}} P(\text{word} \mid \text{topic})^{c_{\text{word}}}$$

(c_{word} = number of times a word appears in the document)

Corpus

Fiction

“the cat in the hat”

“the cow jumped over the moon”

“the cat in the tree”

(total count of all words = 16;

total count of distinct words = 8 (vocabulary))

Non-fiction

“the Giants won the game”

“the candidate won the election”

(total count of all words = 10;

total count of distinct words = 6 (vocabulary))

Prior distributions

$$P(\text{topic} = \text{"fiction"}) = \frac{\text{count of fiction documents}}{\text{total count of documents}} = \frac{3}{5} = .6$$

Conditional distributions

$$P(w \mid t) = \frac{\text{total count of word } w \text{ in all documents of topic } t}{\text{total count of all words in all documents of topic } t}$$

“the cat in the hat”

“the cow jumped over the moon”

“the cat in the tree”

$$P(\text{word} = \text{"the"} \mid \text{fiction}) = \frac{6}{16}$$

$$P(\text{word} = \text{"cat"} \mid \text{fiction}) = \frac{2}{16}$$

$$P(\text{word} = \text{"jumped"} \mid \text{fiction}) = \frac{1}{16}$$

$$P(\text{word} = \text{"over"} \mid \text{fiction}) = \frac{1}{16}$$

$$P(\text{word} = \text{"tree"} \mid \text{fiction}) = \frac{1}{16}$$

“the cat jumped over the tree”

$$\begin{aligned} P(\text{topic} = \text{"fiction"} \mid \text{document} = \text{"the cat jumped over the tree"}) \\ = \\ P(\text{topic} = \text{"fiction"}) \\ \cdot P(\text{word} = \text{"the"} \mid \text{fiction})^2 \\ \cdot P(\text{word} = \text{"cat"} \mid \text{fiction}) \\ \cdot P(\text{word} = \text{"jumped"} \mid \text{fiction}) \\ \cdot P(\text{word} = \text{"over"} \mid \text{fiction}) \\ \cdot P(\text{word} = \text{"tree"} \mid \text{fiction}) \end{aligned}$$

“the cat jumped over the tree” (cont.)

$P(\text{topic} = \text{"fiction"} \mid \text{document} = \text{"the cat jumped over the tree"})$

$$= \frac{3}{5} \times \left(\frac{6}{16}\right)^2 \times \frac{2}{16} \times \frac{1}{16} \times \frac{1}{16} \times \frac{1}{16}$$

$$= \frac{216}{83886080}$$

$$= 2.6 \times 10^{-6}$$

(problem #1: very small number; risk of numerical overflow)

“the cat jumped over the tree” (cont.)

$P(\text{topic} = \text{"non-fiction"} \mid \text{document} = \text{"the cat jumped over the tree"})$

$$\begin{aligned} &= \frac{2}{5} \times \left(\frac{4}{10}\right)^2 \times \frac{0}{10} \times \frac{0}{10} \times \frac{0}{10} \times \frac{0}{10} \\ &= \frac{0}{5000000} \\ &= 0 \end{aligned}$$

(problem #2: unknown words, e.g., “cat” have a 0 conditional probability
→ $P(\text{topic} \mid \text{document}) = 0$)

Log transformation to address problem #1

- Use log probabilities instead:

$$\log P(\text{topic} \mid \text{document}) = \log P(\text{topic}) + \sum_{\text{word}} c_{\text{word}} \log P(\text{word} \mid \text{topic})$$

Laplace (add α , e.g., 1) smoothing to address problem #2

- Add 1 to each word's frequency
- As if we saw each word one more time than we actually did

$$P(w | t) = \frac{(\text{total count of word } w \text{ in all documents of topic } t) + \alpha}{(\text{total count of all words in all documents of topic } t) + \alpha(\text{total count of distinct words in all documents of topic } t)}$$

“the cat jumped over the tree” (take 2)

$$\begin{aligned} & \log P(\text{topic} = \text{"fiction"} \mid \text{document} = \text{"the cat jumped over the tree"}) \\ &= \log \frac{3}{5} + 2 \times \log \frac{6+1}{16+8} + \log \frac{2+1}{16+8} + \log \frac{1+1}{16+8} + \log \frac{1+1}{16+8} + \log \frac{1+1}{16+8} = -12.5 \end{aligned}$$

$$\begin{aligned} & \log P(\text{topic} = \text{"non-fiction"} \mid \text{document} = \text{"the cat jumped over the tree"}) \\ &= \log \frac{2}{5} + 2 \times \log \frac{4+1}{10+6} + \log \frac{0+1}{10+6} + \log \frac{0+1}{10+6} + \log \frac{0+1}{10+6} + \log \frac{0+1}{10+6} = -14.3 \end{aligned}$$



$\operatorname{argmax}_{\text{topic}} \log P(\text{topic} \mid \text{"the cat jumped over the tree"}) = \text{fiction}$

When to use Naïve Bayes?

- Pros
 - Good with “wide data” (i.e., more features p than observations n)
 - Fast to train and predict (also good at online learning, i.e., when new documents are added to the corpus, you just need to increment the word counts)
 - Simple to implement
- Cons
 - Can be hampered by irrelevant features
 - Probabilistic estimates are unreliable because of naïve assumption
 - Often outperformed by other models

Afternoon pairing