UMass CS 585 Intro to NLP
9/14/17 Slides from SLP website

Is this spam?

---------- Forwarded message ----------
From: E-Lotto <infoalert@wp.pl>
Date: Fri, Sep 16, 2016 at 3:24 PM
Subject: Ref#.EL16BDXAUL16-03
To:

Hi,

E-Lotto congratulates you as the winner of $2,500,000.00. Email Rep (Aaron Martins) at aaronmarts@excite.com
with Ref#.EL16BDXAUL20

- Text Classification
- Naive Bayes Model of Documents

$\Longleftrightarrow$

Naive Bayes Classifier

$\Longleftrightarrow$

Class-Conditional LM

Who wrote which Federalist papers?

- 1787-8: anonymous essays try to convince New York to ratify U.S Constitution: Jay, Madison, Hamilton.
- Authorship of 12 of the letters in dispute
- 1963: solved by Mosteller and Wallace using Bayesian methods

James Madison

Alexander Hamilton

Positive or negative movie review?

- unbelievably disappointing

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

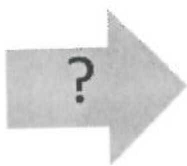
- this is the greatest screwball comedy ever filmed

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

6

What is the subject of this article?

MEDLINE Article



MeSH Subject Category Hierarchy

- Antogonists and Inhibitors
- Blood Supply
- Chemistry
- Drug Therapy
- Embryology
- Epidemiology
- …

7

- Terms of Service interp.
  - Have arb. agr?
- False News vs Real News
- Language ID

Text Classification: definition

- *Input*:
  - a document d → text
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$

J poss. classes

- *Output*: a predicted class $c \in \underline{C}$

Classification Methods: Hand-coded rules

- Rules based on combinations of words or other features
  - spam: black-list-address OR ("dollars" AND"have been selected")
- Accuracy can be high
  - If rules carefully refined by expert
- But building and maintaining these rules is expensive

→ High coverage is hard

↳ Need many rules!

- Human bias

- Alternate spellings
- Problem drift... data distrib. changes

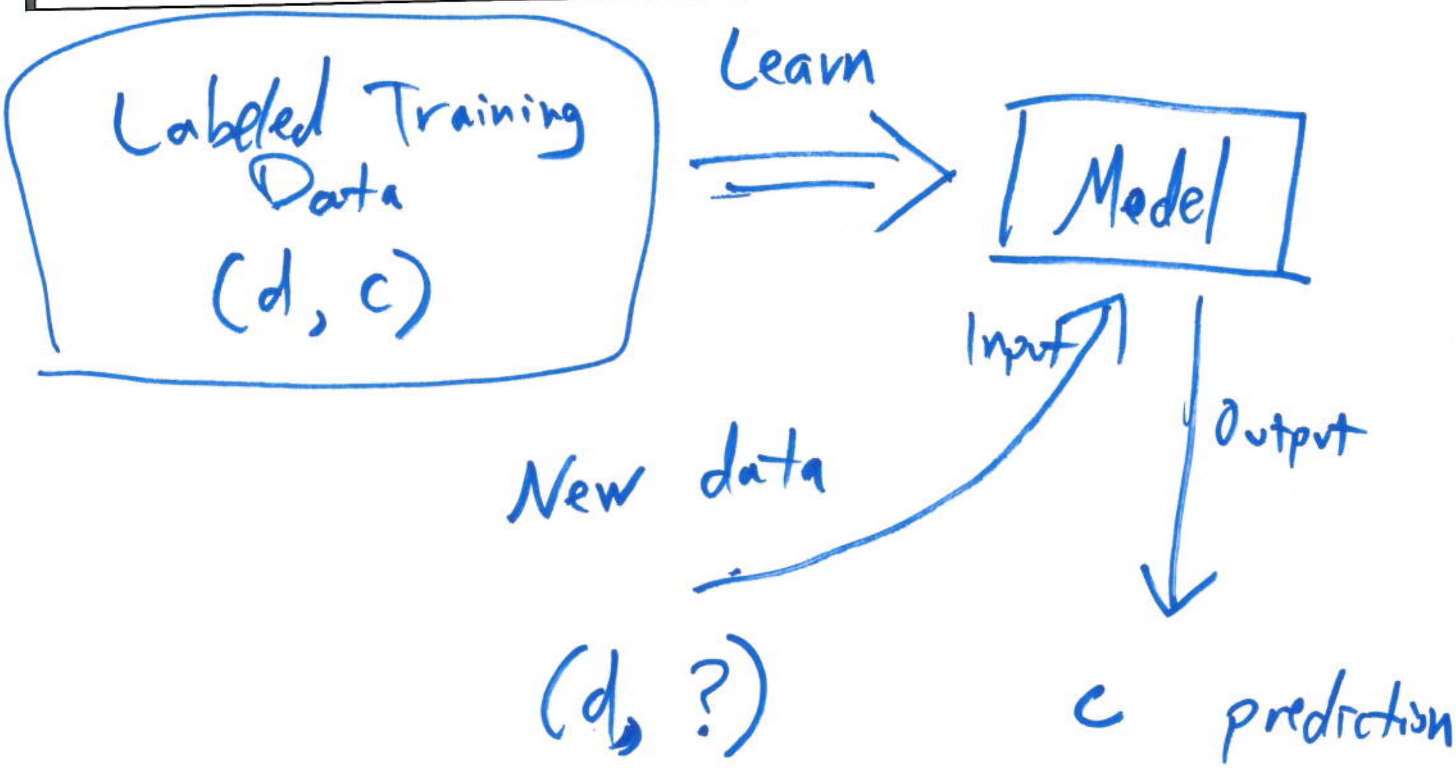
Classification Methods: Supervised Machine Learning

Learning

- *Input:*
  - a document d
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
  - A training set of m hand-labeled documents $(d_1, c_1), \ldots, (d_m, c_m)$
- *Output:*
  - a learned classifier $f: d \rightarrow c$

text Categ.

11



Sup. Learning Algos?

D Trees

K-NN

Linear Regression

Logistic Regression ⇐

SVMs

Neural Networks ≃

Naive Bayes ⇐

The Bag of Words Representation

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
...	...

15

C

C

Problems

- Narrative ordering
  Arg. ordering
- Negations...
  Compositional meaning

Generative Model

doc categ | doc text

$$\begin{tikzcd} c \arrow[r, bend left, "\text{Assume process: LM } P(d|c)"] & d \arrow[l, bend left, "\text{Prediction}"] \end{tikzcd}$$

$$P(c|d) = \frac{\overbrace{P(d|c)}^{\text{Likelihood}}\ \overbrace{P(c)}^{\text{Prior}}}{\underbrace{P(d)}_{\text{Normalizer}}}$$

$$= \sum_{c \in C} P(d|c)P(c)$$

Naïve Bayes Classifier (I)

$$c_{MAP} = \operatorname*{argmax}_{c \in C} P(c \mid d)$$

MAP is "maximum a posteriori" = most likely class

$$= \operatorname*{argmax}_{c \in C} \frac{P(d \mid c) P(c)}{\cancel{P(d)}}$$

Bayes Rule

$$= \operatorname*{argmax}_{c \in C} P(d \mid c) P(c)$$

Dropping the denominator

$$\max_{x \in \{1,2,3\}} x^2 = 9$$

$$\operatorname*{arg\,max}_{x \in \{1,2,3\}} x^2 = 3$$

Multinomial Naïve Bayes Classifier

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c_j) \prod_{x \in X} P(x \mid c)$$

Multinomial Naïve Bayes Independence Assumptions

$$P(d \mid c) = P(x_1, x_2, \ldots, x_n \mid c)$$

- **Bag of Words assumption:** Assume position doesn't matter
- **Conditional Independence:** Assume the feature probabilities $P(x_i \mid c_j)$ are independent given the class c.

$$P(x_1, \ldots, x_n \mid c) = P(x_1 \mid c) \bullet P(x_2 \mid c) \bullet P(x_3 \mid c) \bullet \ldots \bullet P(x_n \mid c)$$

Can estimate!

Applying Multinomial Naive Bayes Classifiers to Text Classification

positions $\leftarrow$ all word positions in test document

$P(c_j)\ P(d \mid c_j)$

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(x_i \mid c_j)$$

Posterior: $P(c|d) \propto P(d|c)\,P(c)$

Naïve Bayes as a Language Model

- Which class assigns the higher probability to s?

$P(I|pos) =$ 0.1; $P(this|pos) =$ 0.01; d

Model pos	
0.1	I
0.1	love
0.01	this
0.05	fun
0.1	film

Model neg	
0.2	I
0.001	love
0.01	this
0.005	fun
0.1	film

$$P(d|pos) = P(I|+)\,P(love|+)\,P(this|+)\,P(fun|+)\,P(film|+)$$
$$0.1 \times 0.1 \times .01 \times .05 \times .1$$

$$P(d|neg) = P(I|-)\,P(love|-)\,P(this|-)\,P(\ldots)\,P(\ldots)$$
$$.2 \quad .001 \quad .01 \quad .005 \quad .1$$

$$P(d|pos) \overset{?}{>} P(d|neg)$$

"Likelihood Ratio"

$$\frac{P(d|pos)}{P(d|neg)} \overset{?}{>} 1$$

Learning the Multinomial Naïve Bayes Model

- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{doccount(C = c_j)}{N_{doc}}$$

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\underbrace{\sum_{w \in V} count(w, c_j)}_{\text{Num tokens of all docs in } c_j}}$$

dog +

Problem with Maximum Likelihood

- What if we have seen no training documents with the word ***fantastic*** and classified in the topic **positive** (***thumbs-up)***?

$$\hat{P}(\text{"fantastic"} \mid \text{positive}) = \frac{\overset{=0}{count(\text{"fantastic"}, \text{positive})}}{\sum_{w \in V} count(w, \text{positive})} = 0$$

- Zero probabilities cannot be conditioned away, no matter the other evidence!

$$c_{MAP} = \text{argmax}_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c)$$

0 !!!! BAD

Laplace (add-1) smoothing for Naïve Bayes

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

$+\alpha$

$+\alpha/|V|$

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

"Pseudo count smoothing"

"Add-α smoothing"

Multinomial Naïve Bayes: Learning

- From training corpus, extract *Vocabulary*
- Calculate $P(c_j)$ terms
  - For each c_j in C do
    $docs_j \leftarrow$ all docs with class $=c_j$

$$P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$$

- Calculate $P(w_k \mid c_j)$ terms
  - $Text_j \leftarrow$ single doc containing all $docs_j$
  - For each word w_k in *Vocabulary*
    $n_k \leftarrow$ # of occurrences of w_k in $Text_j$

$$P(w_k \mid c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha |Vocabulary|}$$

Generative and Discriminative Text Classification with Recurrent Neural Networks

Dani Yogatama, Chris Dyer, Wang Ling, and Phil Blunsom
DeepMind
{dyogatama,cdyer,lingwang,pblunsom}@google.com

Abstract

We empirically characterize the performance of discriminative and generative LSTM models for text classification. We find that although RNN-based generative models are more powerful than their bag-of-words ancestors (e.g., they account for conditional dependencies across words in a document), they have higher

Table 2: Summary of results on the full datasets.

Models	AGNews	Sogou	Yelp Bin	Yelp Full	DBPed	Yahoo
Naïve Bayes	90.0	86.3	86.0	51.4	96.0	68.7
Kneser–Ney Bayes	89.3	94.6	81.8	41.7	95.4	69.3
MLP Naïve Bayes	89.9	76.1	73.6	40.4	87.2	60.6
Discriminative LSTM	92.1	94.9	92.6	59.6	98.7	73.7
Generative LSTM–independent comp.	90.7	93.5	90.0	51.9	94.8	70.5
Generative LSTM–shared comp.	90.6	90.3	88.2	52.7	95.4	69.3
bag of words (Zhang et al., 2015)	88.8	92.9	92.2	58.0	96.6	68.9
fastText (Joulin et al., 2016)	92.5	96.8	95.7	63.9	98.6	72.3
char-CNN (Zhang et al., 2015)	87.2	95.1	94.7	62.0	98.3	71.2
char-CRNN (Xiao & Cho, 2016)	91.4	95.2	94.5	61.8	98.6	71.7
very deep CNN (Conneau et al., 2016)	91.3	96.8	95.7	64.7	98.7	73.4

Summary: Naive Bayes is Not So Naive

- 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
- A good dependable baseline for text classification
  - But we will see other classifiers that give better accuracy