Text Classification

COSC 6336: Natural Language Processing
Spring 2018

UNIVERSITY of HOUSTON

Some content on these slides was adapted from J&M and from Wei Xu

Class Announcements

- ★ Midterm exam now on March 21st.
- ★ Removing Paper presentations from syllabus
- ★ Practical 1 is posted:
  - System submission is March 7th
  - Report is due March 9th

Today's lecture

- ★ Text Classification
  - Example applications
  - Task definition
- ★ Classical approaches to Text Classification
  - Naive Bayes
  - MaxEnt

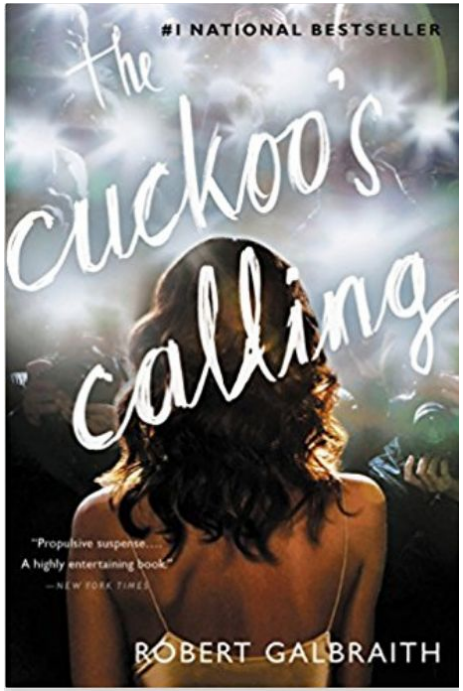
Hello Dear
My name is: Engineer Mrs.Eva Rose, I am a citizen of Austria, a business woman specialized in mining of raw Gold in Africa; but now I am critically sick with esophageal cancer which has damaged almost all the cells in my body system and I will soon die according to my doctors.

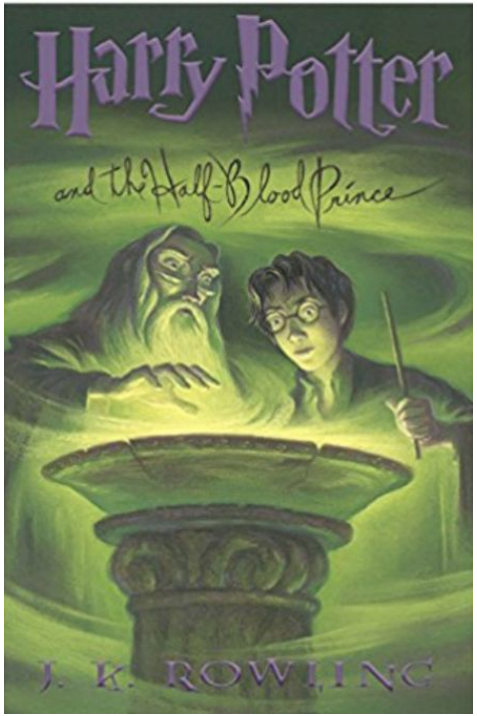
My late husband died in an accident with our two daughters few years ago leaving me with our only son whom is just 10 years old and he is my most concern now as he is still a child and does not know anything about live and has nobody to take care of him after I am dead; because I and my late husband does not have any relatives, we both grew up in the orphanage home and got married under orphanage as orphans. So if I die now my innocent child would be left alone in this wicked world and
I do not wish to send him to any orphanage home, I want him to grow up in the hands of an individual, not orphanage.

Please, i am begging you in the name of God to sincerely accept my proposal; let me instruct my bank to wire transfer my fund worth the sum of US$ 15,000,000.00 (FIFTEEN MILLION DOLLARS) to your account in your country immediately, then you take my son to your home and raise him as your own son. As you receive the fund into your account, you are entitled to take 30 percent and invest 70 percent for my son so that he will not suffer in his entire life.

UNIVERSITY of HOUSTON

What do these books have in common?





Other tasks that can be solved as TC

- ★ Sentiment classification
- ★ Native language identification
- ★ Author Profiling
- ★ Depression detection
- ★ Cyberbullying detection
- ★ ….

Formal definition of the TC task

- Input:
  - a document d
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
- Output: a predicted class $c \in C$

Methods for TC tasks

- ★ Rule based approaches
- ★ Machine Learning algorithms
  - ○ Naive Bayes
  - ○ Support Vector Machines
  - ○ Logistic Regression
  - ○ And now deep learning approaches

Naive Bayes for Text Classification

- ★ Simple approach
- ★ Based on the bag-of-words representation

Bag of words

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!

The first reference to Bag of Words is attributed to a 1954 paper by Zellig Harris

would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

Naive Bayes

Probabilistic classifier $$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d) \quad \text{(eq. 1)}$$

According to Bayes rule: $$P(x|y) = \frac{P(y|x)P(x)}{P(y)} \quad \text{(eq. 2)}$$

Replacing eq. 2 into eq. 1: $$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d) = \underset{c \in C}{\operatorname{argmax}} \frac{P(d|c)P(c)}{P(d)}$$

Dropping the denominator: $$\underset{c \in C}{\operatorname{argmax}} P(d|c)P(c)$$

Naive Bayes

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

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \overbrace{P(f_1, f_2, \ldots, f_n | c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}}$$

How many parameters do we need to learn in this model?

Naive Bayes Assumptions

1. Position doesn't matter
2. Naive Bayes assumption: probabilities $P(f_i|c)$ are independent given the class c and thus we can multiply them:

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

This leads us to:

$$c_{NB} = \underset{c \in C}{\text{argmax}}\, P(c) \prod_{f \in F} P(f|c)$$

Naive Bayes in Practice

We consider word positions:

$$\begin{aligned} \text{positions} &\leftarrow \text{all word positions in test document} \\ c_{NB} &= \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{i \in positions} P(w_i|c) \end{aligned}$$

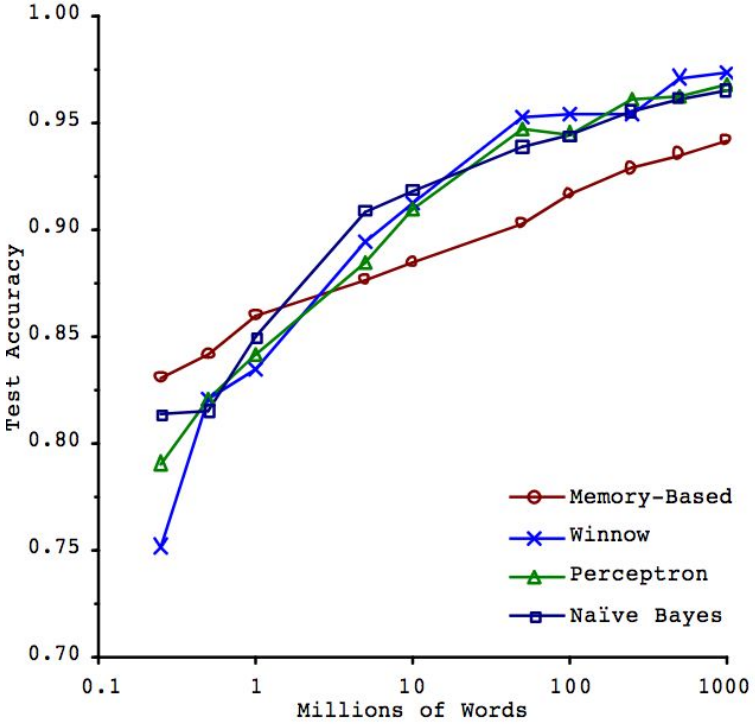
We also do everything in log space:

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} \log P(c) + \sum_{i \in positions} \log P(w_i|c)$$

Naive Bayes: Training

How do we compute $P(c)$ and $P(f_i|c)$?

Is Naive Bayes a good option for TC?



Scaling to Very Very Large Corpora for Natural Language Disambiguation

Michele Banko and Eric Brill
Microsoft Research
1 Microsoft Way
Redmond, WA 98052 USA
{mbanko,brill}@microsoft.com

Abstract

The amount of readily available on-line text has reached hundreds of billions of words and continues to grow. Yet for most core natural language tasks, algorithms continue to be optimized, tested and compared after training on corpora consisting of only one million words or less. In this paper, we potentially large cost of annotating data for those learning methods that rely on labeled text.

The empirical NLP community has put substantial effort into evaluating performance of a large number of machine learning methods over fixed, and relatively small, data sets. Yet since we now have access to significantly more data, one has to wonder what conclusions that have been drawn on small data sets may carry over when these learning methods are trained using much larger corpora.

Evaluation in TC

Confusion table

	Gold Standard	
	True	False
True	TP = true positives	FP = False positives
False	FN = false negatives	TN = True negatives

$$\text{Accuracy} = \frac{TP + TN}{(TP + TN + FN + FP)}$$

Evaluation in TC: Issues with Accuracy?

Suppose we want to learn to classify each message in a web forum as “extremely negative”. We have a collected gold standard data:

- ★ 990 instances are labeled as negative
- ★ 10 instances are labeled as positive
- ★ Test data has 100 instances (99- and 1+)
- ★ A dumb classifier can get 99% accuracy by always predicting “negative” !

More Sensible Metrics: Precision, Recall and F-measure

P= TP/(TP+FP)

R=TP/(TP+FN)

	Gold Standard	
	True	False
True	TP = true positives	FP = False positives
False	FN = false negatives	TN = True negatives

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

UNIVERSITY of HOUSTON

What about Multi-class problems?

- Multi-class: $c > 2$
- P, R, and F-measure are defined for a single class
- We assume classes are mutually exclusive
- We use per class evaluation metrics

$P = \frac{c_{ii}}{\sum_j c_{ji}}$	$R = \frac{c_{ii}}{\sum_j c_{ij}}$

Micro vs Macro Average

- ★ Macro average: measure performance per class and then average
- ★ Micro average: collect predictions for all classes then compute TP, FP, FN, and TN
- ★ Weighted average: compute performance per label and then average where each label score is weighted by its support

Example

Class 1: Urgent

	true urgent	true not
system urgent	8	11
system not	8	340

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

Class 2: Normal

	true normal	true not
system normal	60	55
system not	40	212

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

Class 3: Spam

	true spam	true not
system spam	200	33
system not	51	83

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

Pooled

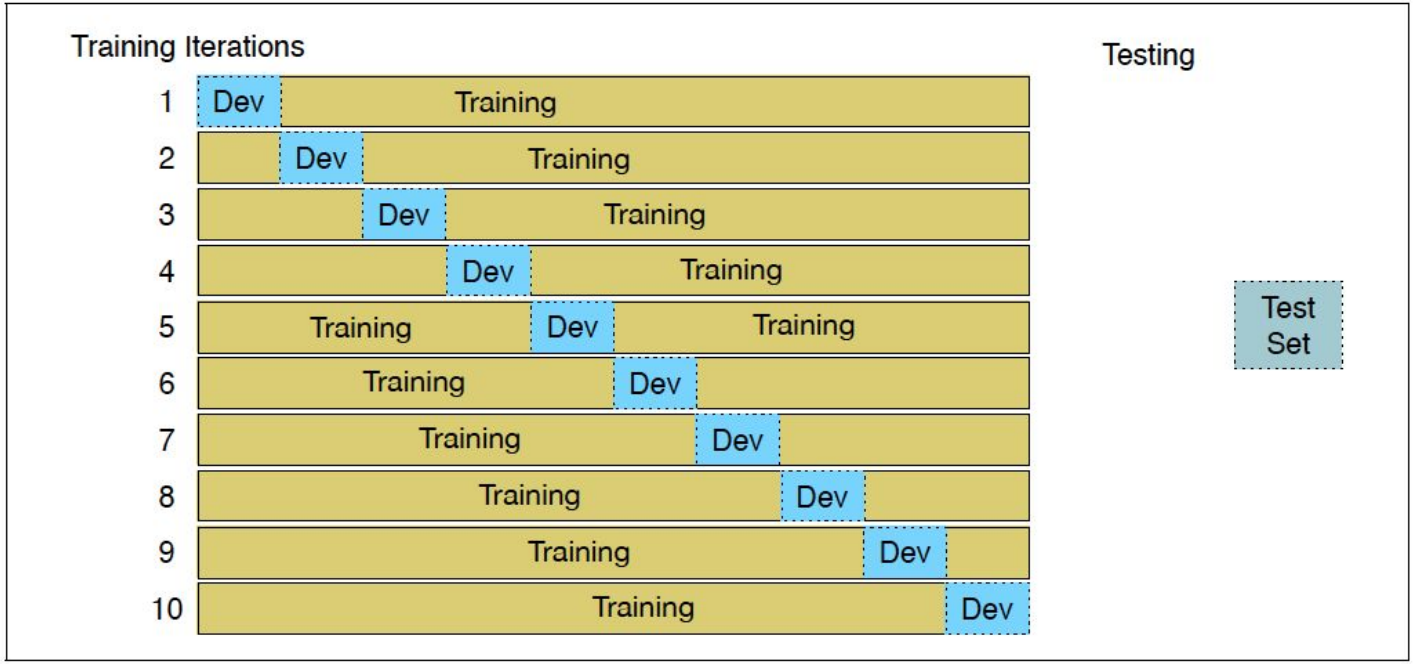
	true yes	true no
system yes	268	99
system no	99	635

$$\text{microaverage precision} = \frac{268}{268+99} = \mathbf{.73}$$

$$\text{macroaverage precision} = \frac{.42+.52+.86}{3} = \mathbf{.60}$$

UNIVERSITY of HOUSTON

Train/Test Data Separation



Does this model look familiar?

w	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
...	...	...

Logistic Regression (MaxEnt)

Disadvantages of NB

- ★ Correlated features
  - Their evidence is overestimated
- ★ This will hurt classifier performance

Logistic Regression

- ★ No independence assumption
- ★ Feature weights are learned simultaneously

Logistic Regression

Computes p(y|x) directly from training data:

$$\sum_{i=0}^{|X|} w_i x_i$$

Logistic Regression

Computes p(y|x) directly from training data:

$$z = \sum_{i=0}^{|X|} w_i x_i$$

But since the result of this will not be a proper probability function we need to normalize this:

$$\frac{e^z}{e^z + 1} = \frac{1}{1 + e^{-z}}$$

Logistic Regression

- ★ We then compute the probability (y|x) for every class
- ★ We choose the class with the highest probability

Features in Text Classification

★ We can have indicator functions of the form:

$$f_1(c,x) = \begin{cases} 1 & \text{if } \text{“great”} \in x \ \& \ c = + \\ 0 & \text{otherwise} \end{cases}$$

$$f_2(c,x) = \begin{cases} 1 & \text{if } \text{“second-rate”} \in x \ \& \ c = - \\ 0 & \text{otherwise} \end{cases}$$

$$f_3(c,x) = \begin{cases} 1 & \text{if } \text{“no”} \in x \ \& \ c = - \\ 0 & \text{otherwise} \end{cases}$$

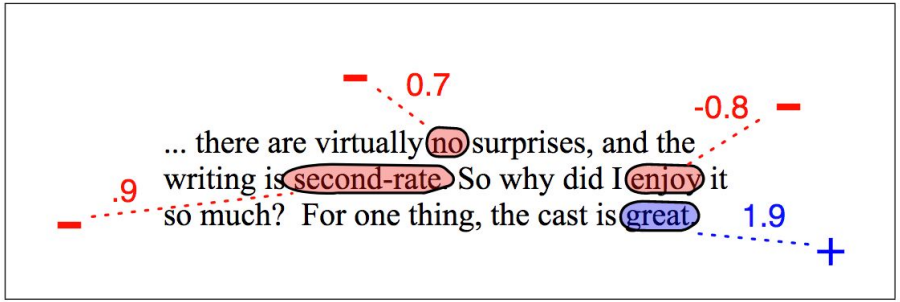
$$f_4(c,x) = \begin{cases} 1 & \text{if } \text{“enjoy”} \in x \ \& \ c = - \\ 0 & \text{otherwise} \end{cases}$$

Features in Text Classification

- ★ Or other functions:

$$f(d) = \begin{pmatrix} \text{count}(\text{“boring”}) \\ \text{count}(\text{“not boring”}) \\ \text{length of document} \\ \text{author of document} \\ \vdots \end{pmatrix}$$

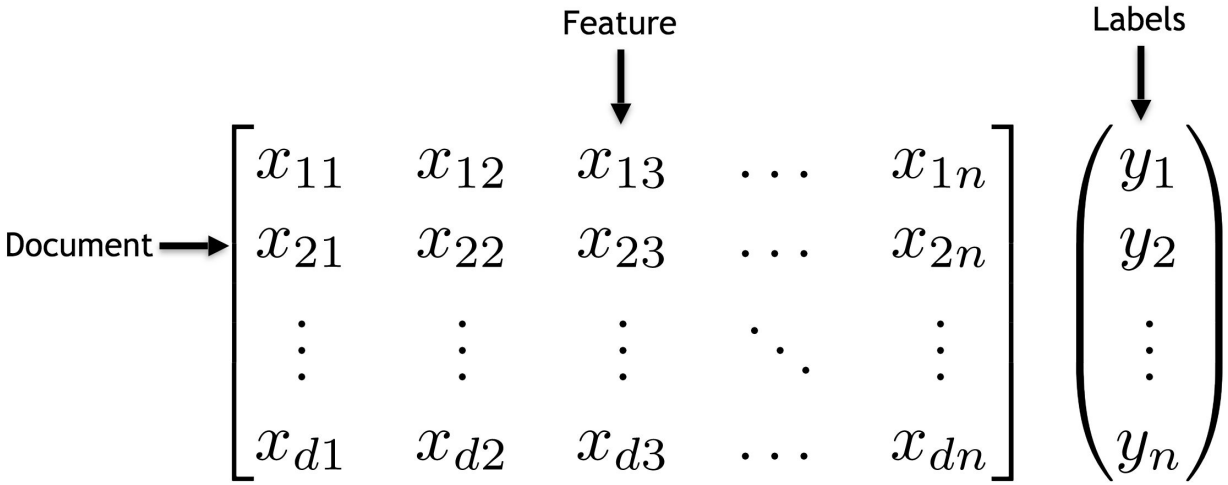
An Example



$$P(+|x) = \frac{e^{1.9}}{e^{1.9} + e^{.9+.7-.8}} = .82$$

$$P(-|x) = \frac{e^{.9+.7-.8}}{e^{1.9} + e^{.9+.7-.8}} = .18$$

Learning in Logistic Regression



Learning in Logistic Regression

How does training look like?

We need to find the right weights

How?

By using the Conditional Maximum Likelihood Principle:

Choose parameters that maximize the log probability of the y labels in the training data:

$$\hat{w} = \underset{w}{\operatorname{argmax}} \sum_{j} \log P(y^{(j)}|x^{(j)})$$

Learning in Logistic Regression

So the objective function L(w) we are maximizing is this:

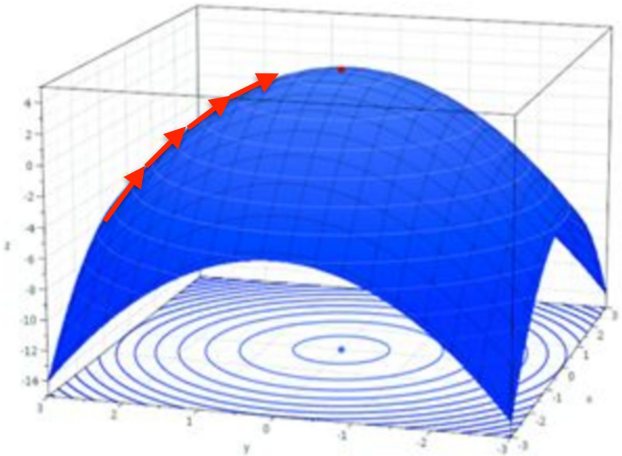
$$\sum_j \log P(y^{(j)}|x^{(j)})$$

$$= \sum_j \log \frac{\exp\left(\sum_{i=1}^{N} w_i f_i(y^{(j)}, x^{(j)})\right)}{\sum_{y' \in Y} \exp\left(\sum_{i=1}^{N} w_i f_i(y'^{(j)}, x^{(j)})\right)}$$

$$= \sum_j \log \exp\left(\sum_{i=1}^{N} w_i f_i(y^{(j)}, x^{(j)})\right) - \sum_j \log \sum_{y' \in Y} \exp\left(\sum_{i=1}^{N} w_i f_i(y'^{(j)}, x^{(j)})\right)$$

Learning in Logistic Regression

- ★ We use optimization methods like Stochastic Gradient Ascent to find the weights that maximize L(w)
- ★ We start with weights = 0 and move in the direction of the gradient (the partial derivative L'(w))



Learning in Logistic Regression

- The derivative turns out to be easy:

$$L'(w) = \sum_j f_k(y^{(j)}, x^{(j)}) - \sum_j \sum_{y' \in Y} P(y'|x^{(j)}) f_k(y'^{(j)}, x^{(j)})$$

Regularization

- ★ Overfitting: when a model “memorizes” the training data
- ★ To prevent overfitting we penalize large weights:

$$\hat{w} = \underset{w}{\operatorname{argmax}} \sum_{j} \log P(y^{(j)}|x^{(j)}) - \alpha R(w)$$

- ★ Regularization forms:
  - L2 norm (Euclidean)

$$\hat{w} = \underset{w}{\operatorname{argmax}} \sum_{j} \log P(y^{(j)}|x^{(j)}) - \alpha \sum_{i=1}^{N} w_i^2$$

  - L1 norm (Manhattan)

$$\hat{w} = \underset{w}{\operatorname{argmax}} \prod_{j}^{M} P(y^{(j)}|x^{(j)}) \times \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(w_i - \mu_j)^2}{2\sigma_j^2}\right)$$

In Summary

- ★ NB is a generative model
  - Highly correlated features can result in bad results
- ★ Logistic Regression
  - Weights for feature are calculated simultaneously
- ★ NB seems to fare better in small data sets
- ★ Both NB and Logistic regression are linear classifiers
- ★ Regularization helps alleviate overfitting