Naive Bayes & Logistc Regression

Natural Language Processing

Lecture 5

THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

Review of Relevant Concepts from Probability.

Definition of Probability

- Relative Frequency
  - $P(sucess) = \lim_{Number\ of\ trials \to \infty} \frac{Number\ of\ sucess}{Number\ of\ trials}$
- Subjective Probability
  - Baysian Statistics
- Axiomatic Probability
  - Set theory, Random Variables, Random Process

Set

- In mathematics, a set is a collection of elements.
- The set with no element is the empty set.
- The elements that make up a set can be any kind of mathematical objects: symbols, numbers, points in space, lines, other geometrical shapes, variables, or even other sets.
- The sample space is the set of all possible outcomes or results of that experiment.
- Example: Set of all capital letters.

Probabilistic Model

- To define a probabilistic model, one must define:

1- A sample space: the set of all possible outcomes of the experiment.

2- A probability law. A probability law assigns a probability, P(A), to every event, A. Must satisfy the Axioms of Probability.

Probability Axioms

1- $P(A) \geq 0$, for every event A.

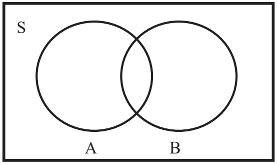
2- If A and B are two disjoint events, then the probability of their union satisfies.

$P(A \cup B) = P(A) + P(B)$

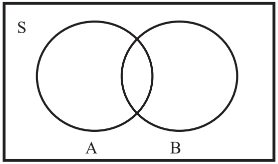
3- The probability of the entire sample space is equal to 1.

$P(S) = 1$

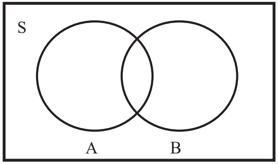
Some Properties of Probability



$A \cup B$



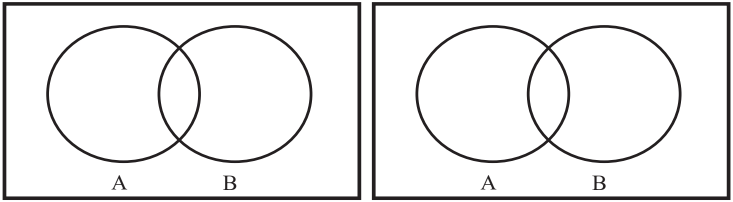
$A \cap B$



$A - B$

- If $A \subset B$ then $P(A) \leq P(B)$
- $P(A\cup)B) = P(A) + P(B) - P(A \cap B)$
- $P(A \cup B) \leq P(A) + P(B)$
- If A and B are disjoint means $P(A \cap B) = \phi$
- If A and B are independent means $P(A \cap B) = P(A) \cdot P(B)$

Properties of Conditional Probability



- The conditional probability of an event$A$, given an event B with $P(B) > 0$, is defined by

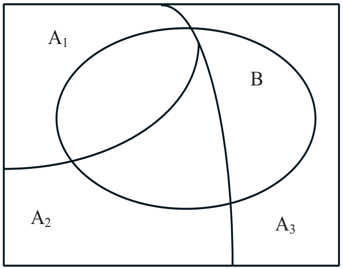
$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

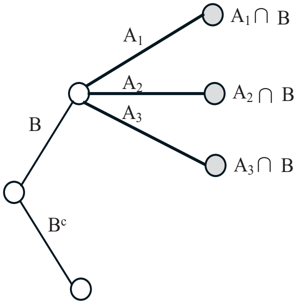
- Conditional probabilities can also be viewed as a probability law on a new universe B, because all of the conditional probability is concentrated on B.

Total Probability Theorem

- Let $A_1, \cdots, A_n$ be disjoint events that form a partition of the sample space (each possible outcome is included in one and only one of the events $A_1, \cdots, A_n$) and assume that $P(A_i) > 0$, for all $i = 1, \cdots, n$. Then, for any event B, we have

$$P(B) = P(A_1)P(B|A_1) + \cdots + P(A_n)P(B|A_n)$$



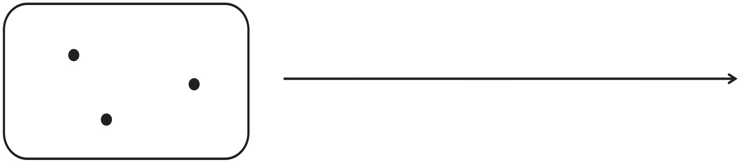


Bayes Rule

- Let $A_1, \cdots, A_n$ be disjoint events that form a partition of the sample space and assume $P(A_i) > 0$, for all i. Then, for any event B such that $P(B) > 0$ we have

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

Random Variable



- Random Variable is a mapping from events to a real line
- Probability Distribution Function (PDF)

$$F_a(\lambda) = P(\{s : A(s) \leq \lambda\})$$

- Example: Draw PDF Probability of flip a coin on a real coordinates.
- $P(\{s : \lambda_1 < A(s) \leq \lambda_2\}) = F_a(\lambda_2) - F_a(\lambda_1)$

Example: Roll Die

Example: Flip Coin

Random Variables and Density Functions

Probability distribution

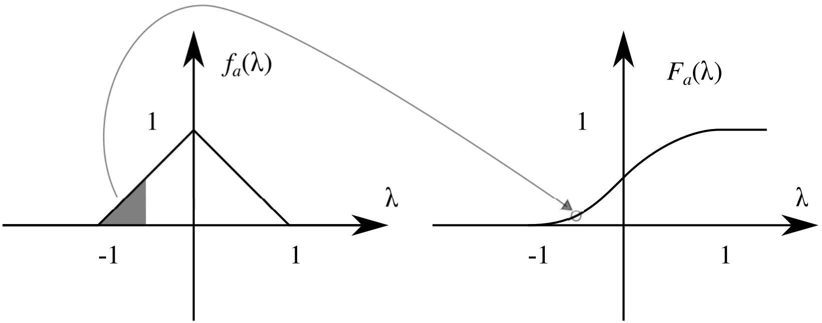
$$F_a(\lambda) = P(a \leq \lambda)$$

Probability density

$$f_a(\lambda) = \frac{\delta F_a(\lambda)}{\delta \lambda}$$

$$F_a(\lambda) = \int_{-\infty}^{\lambda} f_a(\gamma) d\gamma$$

Example density and distribution functions



Example density functions

Gaussian

$$f_a(\lambda) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\lambda-\mu)^2}{2\sigma^2}}$$

Gamma

$$f_a(\lambda) = \frac{1}{\Gamma(k)\theta^k} \lambda^{k-1} e^{-\frac{\lambda}{\theta}}$$

Maxwell-Boltzmann

$$f_a(\lambda) = \sqrt{\left(\frac{m}{2\pi kT}\right)^3} 4\pi\lambda^2 e^{-\frac{m\lambda^2}{2kT}}$$

Joint random variables

Joint density

$$f_{a_1,a_2}(\lambda_1,\lambda_2) = f_{a_1|a_2}(\lambda_1|\lambda_2) f_{a_2}(\lambda_2)$$

Marginal density

$$f_{a_1}(\lambda_1) = \int_{-\infty}^{\infty} f_{a_1,a_2}(\lambda_1,\lambda_2)\, d\lambda_2$$

Conditional density

$$f_{a_2|a_1}(\lambda_2|\lambda_1) = \frac{f_{a_1,a_2}(\lambda_1,\lambda_2)}{f_{a_1}(\lambda_1)}$$

Bayes rule

$$f_{a_1|a_2}(\lambda_1|\lambda_2) = \frac{f_{a_2|a_1}(\lambda_2|\lambda_1) f_{a_1}(\lambda_1)}{f_{a_2}(\lambda_2)}$$

Example: PDF

Estimation

- If some parameters of the density are unknown, we can collect samples of the random variable and estimate the parameters.
- For example, given a set of independent samples from a Gaussian density, we can estimate the mean using the average value.

$$\hat{\mu} = \frac{1}{Q} \sum_{i=1}^{Q} a_i$$

Maximum likelihood estimation

- Assume that the form of the joint density function of the samples is known.
- Select the unknown parameters so that they maximize the density function evaluated at the samples (called the likelihood function).

Joint density

$$f_{a_1,a_2,\ldots,a_Q}(\lambda_1, \lambda_2, \ldots, \lambda_Q; \theta)$$

Likelihood function

$$L(\theta) = f_{a_1,a_2,\ldots,a_Q}(a_1, a_2, \ldots, a_Q; \theta)$$

Maximum likelihood estimate

$$\hat{\theta}_{ML} = \arg\max_{\theta} L(\theta)$$

Maximum likelihood example – exponential density

$$f_a(\lambda;\theta) = \theta e^{-\theta\lambda}$$

$$f_{a_1,a_2,\ldots,a_Q}(\lambda_1,\lambda_2,\ldots,\lambda_Q;\theta) = \theta^Q e^{-\theta\sum_{i=1}^{Q}\lambda_i}$$

$$L(\theta) = f_{a_1,a_2,\ldots,a_Q}(a_1,a_2,\ldots,a_Q;\theta) = \theta^Q e^{-\theta\sum_{i=1}^{N}a_i}$$

$$l(\theta) = \ln(L(\theta)) = Q\ln(\theta) - \theta\sum_{i=1}^{N}a_i$$

$$\frac{dl(\theta)}{d\theta} = \frac{Q}{\theta} - \sum_{i=1}^{Q}a_i = 0$$

$$\hat{\theta}_{ML} = \frac{Q}{\sum_{i=1}^{Q}a_i}$$

ML example – multivariable

$$\begin{aligned}
t_i &= wp_i + v_i \\
f_{v_i}(\lambda_i;\theta) &= \theta e^{-\theta\lambda}, \lambda \geq 0 \\
f_{t_i}(\gamma_i;\theta,w) &= \theta e^{-\theta(\gamma_i - wp_i)}, \gamma_i \geq wp_i \\
L(\theta,w) &= \theta^N e^{-\theta\sum_{i=1}^{N}(t_i - wp_i)}, w \leq \frac{t_i}{p_i} \\
l(\theta,w) &= \ln(L(\theta)) = N\ln(\theta) - \theta\sum_{i=1}^{N}(t_i - wp_i), w \leq \frac{t_i}{p_i} \\
\frac{\partial l(\theta,w)}{\partial\theta} &= \frac{N}{\theta} - \sum_{i=1}^{N}(t_i - wp_i) \\
\frac{\partial l(\theta,w)}{\partial w} &= \sum_{i=1}^{N} p_i
\end{aligned}$$

Bayesian estimation

- If we consider θ to be a random variable, rather than an unknown constant, the likelihood function is reinterpreted as a conditional density.

$$f_{a_1,a_2,\ldots,a_Q}(\lambda_1,\lambda_2,\ldots,\lambda_Q;\theta) = f_{\mathbf{a}}(\boldsymbol{\lambda};\theta) \Rightarrow f_{\mathbf{a}|\theta}(\boldsymbol{\lambda} \mid \gamma)$$

- We can then use Bayes rule to compute the posterior density by multiplying the likelihood function by the prior density for θ and normalizing.

$$f_{\theta|\mathbf{a}}(\gamma \mid \boldsymbol{\lambda}) = \frac{f_{\mathbf{a}|\theta}(\boldsymbol{\lambda} \mid \gamma) f_{\theta}(\gamma)}{f_{\mathbf{a}}(\boldsymbol{\lambda})}$$

- We can then estimate θ by either maximizing the posterior density, or computing its expectation (conditional mean), which minimizes mean square error.

Example of Bayesian estimation

- Consider the problem of a signal plus noise.

$$a = \theta + e$$

- Assume that the noise e is Gaussian with mean zero and variance σ^2 and that the prior density of the parameter θ is Gaussian with mean zero and variance σ_θ^2.

$$f_e(\lambda) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\lambda^2}{2\sigma^2}}, f_\theta(\gamma) = \frac{1}{\sqrt{2\pi}\sigma_\theta} e^{-\frac{\gamma^2}{2\sigma_\theta^2}}$$

- The density for a, given knowledge of θ would be

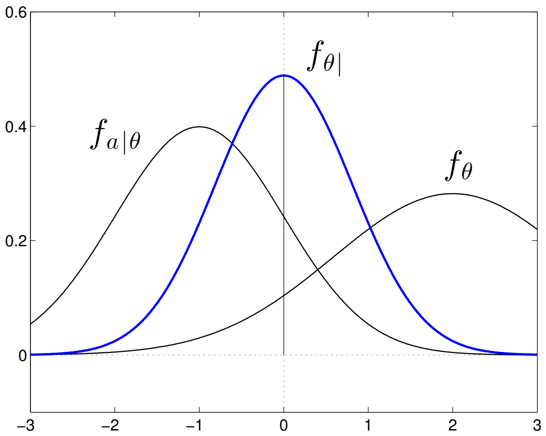
$$f_{a|\theta}(\lambda \mid \gamma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\lambda-\gamma)^2}{2\sigma^2}}$$

- Using Bayes rule, the posterior density would be

$$f_{\theta|a}(\gamma \mid \lambda) = \frac{f_{a|\theta}(\lambda \mid \gamma) f_\theta(\gamma)}{f_a(\lambda)}$$

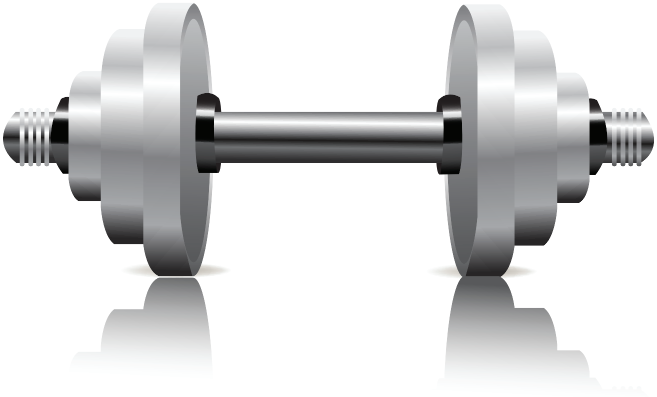
Example

Density functions



Exercise - Lecture 5

- Exercise 1 to 2
- Class-Ex-Lecture5.py



Naive Bayes

Classification

- Classification lies at the heart of both human and machine intelligence
- Deciding what letter, word, or image has been presented to our senses, sorting mail, assigning grades to homeworks these are all examples of assigning a category to an input.
- Many language processing tasks involve classification.
- Text categorization and sentiment analysis are major tasks in NLP.
- Sentiment analysis, the extraction of sentiment, the positive or negative orientation that a writer expresses toward some object.

Task of Text Classification

- Is this email spam?
- Who wrote which Federalist papers?
- What is the subject of this medical article?
- Positive or negative movie review?

Positive or negative movie review?

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

Positive or negative movie review?

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

Why sentiment analysis?

- **Movie**: is this review positive or negative?
- **Products**: what do people think about the new iPhone?
- **Public** sentiment: how is consumer confidence?
- **Politics**: what do people think about this candidate or issue?
- **Prediction**: predict election outcomes or market trends from sentiment

Scherer Typology of Affective States

- Emotion
  - Brief, organically synchronized.. evaluation of a major event => angry, sad, joyful, fearful, ashamed, proud, elated
  - * Ex. Detecting annoyed callers. Ex.2 : Detecting confused Vs Confident students.
- Mood
  - diffuse non-caused low-intensity long-duration change in subjective feeling => cheerful, gloomy, irritable, listless, depressed, buoyant
  - * Ex. Finding depressed writers from their blogs
- Interpersonal Stances
  - Affective stance towards another person in a specific interaction => friendly, flirtatious, distant, cold, warm, supportive, contemtuous
  - * Ex. Detecting flirt / friendliness in talk. We can build an app to check if a girl is perfect for speed dating by just scanning her conversations
- Attitudes
  - Enduring, affectively colored beliefs, disposition towards objects or persons => liking, loving, hating, valuing, desiring.
  - * Ex. Sentiment analysis — whole blog ahead has its information!
- Personality Traits
  - Stable personality dispositions and typical behavior tendencies =¿ nervous, anxious, reckless, morose, hostile, jealous
  - * Ex. Detect Extroverts/Introverts based on conversations.

Basic Sentiment Classification

- Sentiment analysis is the detection of attitudes.
- Is the attitude of this text positive or negative?

Summary: Text Classification

- **Sentiment analysis**
- **Spam detection**
- **Authorship identification**
- **Language Identification**
- **Assigning subject categories, topics, or genres**

Text Classification

- **Input**:

  - a document d

  - a fixed set of classes $C = \{c_1, c_2, ..., c_n\}$

- **Output**:

  - a predicted class $c \in C$

Classification Methods: Supervised Machine Learning

- Naïve Bayes
- Logistic regression
- Neural networks
- k-Nearest Neighbors

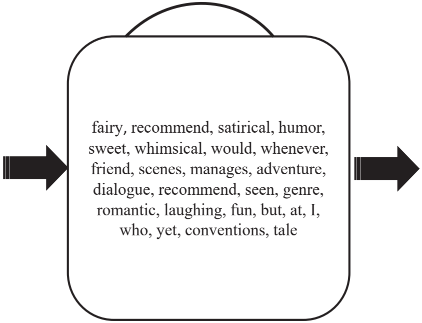
Naive Bayes Intuition

- Simple ("naive") classification method based on Bayes rule

- Relies on very simple representation of document Bag of words.

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 seen, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



fairy, recommend, satirical, humor, sweet, whimsical, would, whenever, friend, scenes, manages, adventure, dialogue, recommend, seen, genre, romantic, laughing, fun, but, at, I, who, yet, conventions, tale

Words	Count
It	6
I	5
the	4
to	3
and	3
seen	2
yet	1
whimsical	1
sweet	1
.	.
.	.
.	.

The Bag of Words Representation

Function (

Words	Count
It	6
I	5
the	4
to	3
and	3
seen	2
yet	1
whimsical	1
sweet	1
.	.
.	.
.	.

) = Class

Input

Bayes Rule Applied to Documents and Classes

- For a document d and a class c

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

- MAP is “maximum a posteriori” = most likely class

$$c_{MAP} = \arg\max_{c \in C} P(c|d)$$

- Bayes Rule

$$c_{MAP} = \arg\max_{c \in C} \frac{P(d|c)P(c)}{P(d)}$$

- Dropping the denominator

$$c_{MAP} = \arg\max_{c \in C} P(d|c)P(c)$$

Bayes Rule Applied to Documents and Classes

- Likelihood and Prior

$$c_{MAP} = \arg\max_{c \in C} P(d|c)P(c)$$

- Document d represented as features Features $x_1, x_2, ..., x_n$

$$c_{MAP} = \arg\max_{c \in C} P(x_1, x_2, ..., x_n|c)P(c)$$

- We can just count the relative frequencies in a corpus.

Naive Bayes Independence Assumptions

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

$$P(x_1, x_2, ..., x_n|c) = P(x_1|c) \cdot P(x_2|c) \cdot P(x_3|c) \cdot, ..., \cdot P(x_n|c)$$

- Maximize

$$c_{MAP} = \arg\max_{c \in C} P(x_1, x_2, ..., x_n|c)P(c)$$
$$c_{MAP} = \arg\max_{c \in C} P(c)\Pi_{x \in X} P(x|c)$$

- Multiplying lots of probabilities can result in floating-point underflow.
- Use logs, because $log(ab) = log(a) + log(b)$; We'll sum logs of probabilities instead of multiplying probabilities

Log Space

- Instead of doing multiplication

$$c_{NB} = \arg\max_{c_j \in C} P(c)\Pi_{i \in pos} P(x_i|c_j)$$

- Log space calculation

$$c_{NB} = \arg\max_{c_j \in C} [\log P(c) + \sum_{i \in pos} \log P(x_i|c_j)]$$

- Taking log doesn’t change the ranking of classes.
- It’s a linear model. Just a max of a sum of weights: a linear function of the inputs So naive bayes is a linear classifier.

Learning the Naive Bayes Model

- First attempt: maximum likelihood estimates.

$$\hat{P(c_j)} = \frac{N_{c_j}}{N_{total}}$$

- Simply use the frequencies in the data

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

- Fraction of times word w_i appears among all words in documents of topic c_j
- Create mega-document for topic j by concatenating all docs in this topic. Use frequency of w in mega-document

Problem with Maximum Likelihood

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

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

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

$$c_{MAP} = \arg\max_{c} \hat{P}(c)\Pi_i \hat{P}(x_i|c)$$

- **Laplace (add-1) Smoothing for Naïve Bayes**

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

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

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 $= c_j$
  ```

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

- Calculate $P(w_k|c_j)$ terms
  - $Text_j \rightarrow$ single doc containing all $docs_j$
  - For each word w_k in Vocabulary

  ```
      $n_k \leftarrow$ of occurrences of $w_k$ in $Text_j$
  ```

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

Unknown words

- What about unknown words that appear in our test data but not in our training data or vocabulary?

∗ We ignore them; Remove them from the test document; Pretend they weren't there; Don't include any probability for them at all!

- Why don't we build an unknown word model?

∗ It doesn't help: knowing which class has more unknown words is not generally helpful

Stop words

- Some systems ignore stop words.
  - * Stop words: very frequent words like the and a.
  - * Sort the vocabulary by word frequency in training set
  - * Call the top 10 or 50 words the stopword list.
  - * Remove all stop words from both training and test sets, as if they were never there
- But removing stop words doesn't usually help
  - * So in practice most NB algorithms use all words and don't use stopword lists

A worked sentiment example with add-1 smoothing

	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	?	predictable ~~with~~ no fun

1. Prior from training:

$$\hat{P}(c_j) = \frac{N_{c_j}}{N_{total}}$$

P(-) = 3/5
P(+) = 2/5

2. Drop "with"

3. Likelihoods from training:

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

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

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

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

4. Scoring the test set:

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

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

Optimizing for Sentiment Analysis

- For tasks like sentiment, word occurrence seems to be more important than word frequency.

* The occurrence of the word fantastic tells us a lot

* The fact that it occurs 5 times may not tell us much more.

- Binary NB

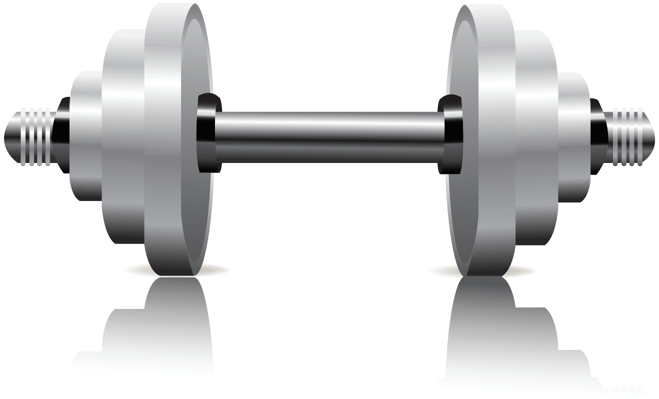
* Clip our word counts at 1.

Sentiment Classification: Dealing with Negation

- Negation changes the meaning of ”like” to negative.
  - I really like this movie
  - I really don’t like this movie
- Negation can also change negative to positive-ish
  - Don’t dismiss this film
  - Doesn’t let us get bored
- Add NOT_ to every word between negation and following punctuation:
  - didn’t like this movie , but I
  - didn’t NOT_like NOT_this NOT_movie but I

Exercise - Lecture 5

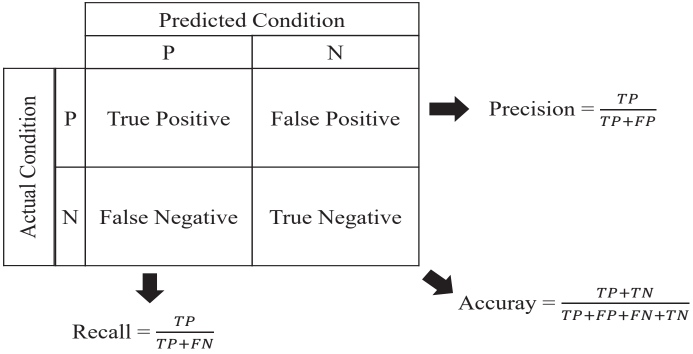
- Exercise 3 to 4
- Class-Ex-Lecture5.py



Evaluation Metric

Confusion Matrix

- Evaluation of the performance of a classification model is based on the counts of test records correctly and incorrectly predicted by the model.



		Predicted Condition	
		P	N
Actual Condition	P	True Positive	False Positive
	N	False Negative	True Negative

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{Accuray} = \frac{TP+TN}{TP+FP+FN+TN}$$

Evaluation: Accuracy

- Why don’t we use accuracy as our metric?
- Imagine we saw 1 million tweets
  - 100 of them talked about Delicious Pie Co.
  - 999,900 talked about something else
- We could build a dumb classifier that just labels every tweet ”not about pie”
  - It would get 99.99% accuracy!!! Wow!!!!
  - But useless! Doesn’t return the comments we are looking for
  - That’s why we use precision and recall instead

Evaluation: Precision and Recall

- Precision is to measure the quality of our predictions only based on what our predictor claims to be positive (regardless of all it might miss)
- However, Recall is to measure such quality with respect to the mistakes we did (what should have been predicted as positive but we flagged as negative)
- Precision and recall, unlike accuracy, emphasize true positives
- F Measure a single number the combines Precision and Recall.

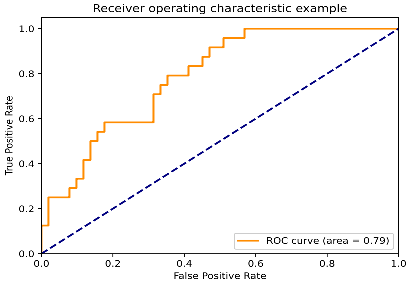
$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

- If $\beta = 1$ we used F_1 score.

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

ROC and AUC Curve

- The receiver operator characteristic is another common tool used for evaluation.
- It plots out the sensitivity and specificity for every possible decision rule cutoff between 0 and 1 for a model.
- For each possible threshold, the ROC curve plots the False positive rate versus the true positive rate.
  - So amongst all the probabilities arranged in ascending order, everything below bad is considered as negative and everything above 0.1 is considered as positive.



Cohen's kappa

- Cohen's kappa: a statistic that measures inter-annotator agreement.
- This function computes Cohen's kappa, a score that expresses the level of agreement between two annotators on a classification problem.
- It is defined as

$$\kappa = \frac{(p_o - p_e)}{1 - p_e}$$

- where p_o is the empirical probability of agreement on the label assigned to any sample (the observed agreement ratio).
- p_e is the expected agreement when both annotators assign labels randomly. p_e is estimated using a per-annotator empirical prior over the class labels

Matthews Correlation

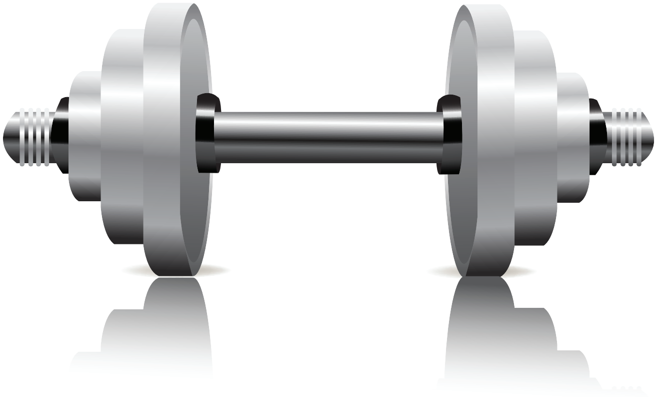
- The Matthews correlation coefficient is used in machine learning as a measure of the quality of binary (two-class) classifications.
- It takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes.
- It takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

- When there are more than two labels, the value of the MCC will no longer range between -1 and +1. Instead the minimum value will be somewhere between -1 and 0 depending on the number and distribution of ground true labels. The maximum value is always +1

Exercise - Lecture 5

- Exercise 5 to 5
- Class-Ex-Lecture5.py



Logistic Regression

Logistic Regression

- Important analytic tool in natural and social sciences.
- Baseline supervised machine learning tool for classification.
- Is also the foundation of neural networks.

- Naive Bayes is a generative classifier.
- Logistic regression is a discriminative classifier.

Generative Classifiers

- Suppose we're distinguishing cat from dog images
  - Build a model of what's in a cat image
  - Knows about whiskers, ears, eyes
  - Assigns a probability to any image.
  - Also build a model for dog images.
- Now given a new image: Run both models and see which one fits better.

Discriminative Classifier

- Just try to distinguish dogs from cats
- Oh look, dogs have collars!
- Let's ignore everything else
- Naive Bayes : likelihood and prior

$$\hat{c} = \arg\max_{c \in C} P(d|c)P(c)$$

- Logistic Regression : Posterrior

$$\hat{c} = \arg\max_{c \in C} P(c|d)$$

Classification in Logistic Regression

- Given a series of input/output pairs: x^i, y^i
- For each observation x^i

- We represent x^i by a feature vector $[x_1, x_2, ..., x_n]$

- We compute an output: a predicted class $\hat{y}^i \in \{0, 1\}$

- For feature x_i, weight w_i tell is how important is x_i
- x_i =”review contains ’awesome’”: $w_i = +10$
- x_j =”review contains ’abysmal’”: $w_j = -10$
- x_k =”review contains ’mediocre’”: $w_k = -2$

Logistic Regression for one observation

- Input observation: vector $x = [x_1, x_1, ..., x_n]$
- Weights: one per feature: $W = [w_1, w_2, ..., w_n]$
- Output: a predicted class $\hat{y} \in \{0, 1\}$

- For each feature x_i, weight w_i tells us important of x_i
- We will sum up all the weighted features and the bias.

$$Z = (\sum_{i=1}^{n} w_i x_i) + b$$

$$Z = w \cdot x + b$$

Probabilistic Classifier

- We need to formalize “sum is high”.
- We want a model that can tell us:

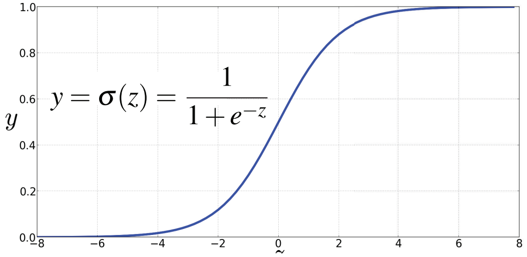
$$p(y = 1|x;\theta)p(y = 0|x;\theta$$

- The problem is not a probability , it is just a number.
  $z = w \cdot x + b$
- Solution : use a function of z that goes from 0 to 1

$$y = \sigma(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + exp^{-z}}$$

Sigmoid or Logistic Function

- We'll compute $w \cdot x + b$ then we'll pass it through the sigmoid function
- And we'll just treat it as a probability



Making Probabilities

$$P(y=1) = \sigma(w \cdot x + b) = \frac{1}{1 + exp(-(w \cdot x + b))}$$

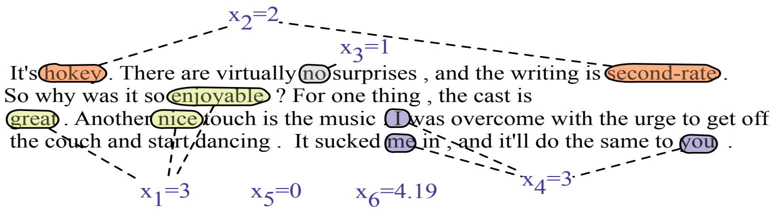
$$P(y=0) = 1 - \sigma(w \cdot x + b) = 1 - \frac{1}{1 + exp(-(w \cdot x + b))}$$

$$P(y=0) = 1 - \sigma(w \cdot x + b) = \frac{exp(-(w \cdot x + b)}{1 + exp(-(w \cdot x + b))}$$

- Turning probability into classifier

$$\hat{y} = 1 if P(y = 1|x) > 0.5 otherwise 0$$

Sentiment Example 1



$x_2=2$
$x_3=1$
It's hokey . There are virtually no surprises , and the writing is second-rate .
So why was it so enjoyable ? For one thing , the cast is
great . Another nice touch is the music . I was overcome with the urge to get off
the couch and start dancing . It sucked me in , and it'll do the same to you .
$x_1=3$
$x_5=0$
$x_6=4.19$
$x_4=3$

Var	Definition	Value
x_1	count(positive lexicon) $\in$ doc)	3
x_2	count(negative lexicon) $\in$ doc)	2
x_3	$\begin{cases} 1 & \text{if “no”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if “!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	log(word count of doc)	$\ln(66) = 4.19$

Sentiment Example 2

- Suppose $W = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7]$

$$\begin{aligned} p(+|x) = P(Y=1|x) &= \sigma(w \cdot x + b) \\ &= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1) \\ &= \sigma(.833) \\ &= 0.70 \end{aligned}$$

$$\begin{aligned} p(-|x) = P(Y=0|x) &= 1 - \sigma(w \cdot x + b) \\ &= 0.30 \end{aligned}$$

Learning Weights

- where did the W's come from?
- We need a distance estimator: a loss function or a cost function.
- We need an optimization algorithm to update w and b to minimize the loss.
- A loss function: **cross-entropy loss**
- An optimization algorithm: **stochastic gradient descent**

Deriving Cross-Entropy Loss

- Goal : Maximize probability of the correct label $p(y|x)$ since there are only 2 discrete outcomes (0,1).

$$p(y|x) = \hat{y}^y (1-\hat{y})^(1-y)$$

If y=1, this simplifies to $\hat{y}$

If y=0, this simplifies to $1-\hat{y}$

$$\log p(y|x) = \log[\hat{y}^y (1-\hat{y})^(1-y)]$$
$$\log p(y|x) = y \log \hat{y} + (1-y)\log(1-\hat{y})$$

Deriving Cross-Entropy Loss

- Goal : Maximize probability of the correct label $p(y|x)$

$$\log p(y|x) = y \log \hat{y} + (1 - y) \log(1 - \hat{y})$$

- Now flip the sign to turn this into a loss: something to minimize
- Cross-entropy loss

$$L_{CE}(y, \hat{y}) = -\log p(y|x) = -[y \log \hat{y} + (1 - y) \log(1 - \hat{y})]$$
$$L_{CE}(y, \hat{y}) = -[y \log \sigma(w \cdot x + b) + (1 - y) \log(1 - \sigma(w \cdot x + b))]$$

- We want loss to be: smaller if the model estimate is close to correct

Exercise - Lecture 5

- Exercise 6 to 7
- Class-Ex-Lecture5.py