CS 4650/7650: Natural Language Processing

Text Classification

Diyi Yang

Some slides borrowed from Jacob Eisenstein (was at GT) and Dan Jurafsky at Stanford
TA Office Hours

- Ian Stewart: Tuesdays, 2-4pm, Coda C1106
- Jiaao Chen: Thursdays, 2-4pm, Coda C1008
- Nihal Singh: Fridays, 9-11am, Coda C1008
- Jingfeng Yang: Mondays, 10am-12pm, Coda 14th common area
Sign Up for Piazza

https://piazza.com/gatech/spring2020/cs7650cs4650/home
Staff Mailing List

cs4650-7650-s20-staff@googlegroups.com
Waiting List
Your Homework 1

- **Due date:** Jan 15\(^{th}\), 3:00pm, EST
Other Questions?
Very Quick Review on Probabilities

- Event space (e.g., $X, Y$) – in this class, usually discrete
- Random variables (e.g., $X, Y$)
- Random variable $X$ takes value $x, x \in X$ with probability $p(X = x)$ or $p(x)$
Very Quick Review on Probabilities

- Joint probability $p(X = x, Y = y)$
- Conditional probability $p(X = x | Y = y) = \frac{p(X=x,Y=y)}{p(Y=y)}$
Very Quick Review on Probabilities

- **Always true:**
  
  \[ p(X = x, Y = y) = p(X = x|Y = y) \cdot p(Y = y) = P(Y = y|X = x) \cdot P(X = x) \]

- **Sometimes true:**
  
  \[ p(X = x, Y = y) = p(X = x) \cdot p(Y = y) \]
The number of ways to select $k$ words out of $n$ given words ("unordered samples without replacement")

$$\binom{n}{k} = \frac{n!}{n!(n-k)!}$$

- Here, $n, n_1, n_2, \ldots, n_k$ are all non-negative integers, and $n_1 + n_2 + n_3 + \cdots n_k = n$
- The number of ways to split $n$ distinct words into $k$ distinct groups of sizes $n_1, \ldots, n_k$, respectively
Classification

- A mapping $h$ from input data $x$ (drawn from instance space $\mathcal{X}$) to a label $y$ from some enumerable output space $\mathcal{Y}$
  - $\mathcal{X} =$ set of all documents
  - $\mathcal{Y} =$ \{English, Mandarin, Greek, …\}
  - $x =$ a single document
  - $y =$ ancient Greek
Movie Ratings

positive

“... is a film which still causes real, not figurative, chills to run along my spine, and it is certainly the bravest and most ambitious fruit of Coppola’s genius”

Roger Ebert, Apocalypse Now

negative

• “I hated this movie. Hated hated hated hated hated this movie. Hated it. Hated every simpering stupid vacant audience-insulting moment of it. Hated the sensibility that thought anyone would like it.”

Roger Ebert, North
Customer Review

**NOT DISHWASHER SAFE**
Reviewed in the United States on April 5, 2019
Color: Blue  Verified Purchase

Used the bottle for one day. There was a slight lid leak, but I was willing to overlook that because I liked the other aspects of the product. Put it in the dishwasher with my other water bottles, air dry, and it melted. There is nothing in the product description that indicates it is not dishwasher safe, nor was there a product sheet included with the bottle indicating to hand wash only. I have a number of plastic water bottles that I routinely send through the dishwasher on this setting and have never had a problem. Extremely disappointed!

19 people found this helpful

**Makes Drinking Water Fun**
Reviewed in the United States on March 31, 2019
Color: Transparent  Verified Purchase

It is always a challenge to drink the recommended amount of water each day, so important for health. This bottle makes it fun while serving as a reminder to keep drinking! Bottle is good quality, handle makes it easy to lift.

14 people found this helpful

Customer reviews

4.5 out of 5

451 customer ratings

- 5 star: 78%
- 4 star: 9%
- 3 star: 5%
- 2 star: 2%
- 1 star: 6%

By feature

- Sturdiness: 4.5
- Flavor: 4.5
- Durability: 4.4
Political Opinion Mining

**emilia @PoliticalEmilia · 43m**
As somebody whose immediate family are immigrants from Iran, I want to remind that this isn’t the fault of Iranian Americans. Most of us want no more war in the Middle East.

Take your anger out at your government leaders, not at us. We have nothing to do with it. #IranAttacks

💬 81  🔄 239  ❤️ 1.9K

**Nithya Raman @nithyavrman · Jan 6**
LA is one of the most immigrant-rich cities in the US.

Almost 50% of residents are foreign-born. 10% are undocumented.

As Trump works to implement his racist agenda, what are our elected officials doing to defend immigrant Angelenos?

💬 55  🔄 138  ❤️ 606

**Brigitte Gabriel @ACTBrigitte · 3m**
Thank Goodness there were ZERO U.S. casualties from the attacks Iran made tonight.

President Trump is monitoring the situation with his top leaders right now.

I’ve never felt more comfortable with a leader at the helm, than I do tonight with President Trump in office.

💬 21  🔄 145  ❤️ 413

**Palmer Report @PalmerReport · 1m**
So a foreign nation fired missiles at U.S. troops tonight, and the President of the United States ISN’T addressing the nation? How far gone is Donald Trump? His handlers don’t even trust him to read a speech off a teleprompter anymore.

💬 15  🔄 74  ❤️ 225

**Andrea Chalupa @AndreaChalupa · 7m**
Trump is betting on Iran doing something so horrific to Americans that we rally around the flag, and the 2020 election becomes a mindless debate of who’s "patriotic" vs. who’s anti-war ("weak" on Iran).
Female or Male Author?

1. By 1925 present-day Vietnam was divided into three parts under French colonial rule. The southern region embracing Saigon and the Mekong delta was the colony of Cochin-China; the central area with its imperial capital at Hue was the protectorate of Annam...

2. Clara never failed to be astonished by the extraordinary felicity of her own name. She found it hard to trust herself to the mercy of fate, which had managed over the years to convert her greatest shame into one of her greatest assets...

Is This Spam?

Subject: Important notice!
From: Stanford University <newsforum@stanford.edu>
Date: October 28, 2011 12:34:16 PM PDT
To: undisclosed-recipients;;

Greats News!

You can now access the latest news by using the link below to login to Stanford University News Forum.


Click on the above link to login for more information about this new exciting forum. You can also copy the above link to your browser bar and login for more information about the new services.

© Stanford University. All Rights Reserved.
What Is the Subject of This Article?

MEDLINE Article

MeSH Subject Category Hierarchy

- Antagonists and Inhibitors
- Blood Supply
- Chemistry
- Drug Therapy
- Embryology
- Epidemiology
- ...

?
This Class

- Basic representations of text data for classification
- Three linear classifiers
  - Naïve Bayes
  - Perception
  - Logistic regression
The Text Classification Problem

- Given a text $w = (w_1, w_2, ..., w_T) \in \mathcal{V}^*$, predict a label $y \in \mathcal{Y}$
# Some Direct Text Classification Applications

<table>
<thead>
<tr>
<th>Task</th>
<th>$x$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language identification</td>
<td>text</td>
<td>{English, Mandarin, Greek, … }</td>
</tr>
<tr>
<td>Spam classification</td>
<td>email</td>
<td>{spam, not spam}</td>
</tr>
<tr>
<td>Authorship attribution</td>
<td>text</td>
<td>{jk rowling, james joyce, … }</td>
</tr>
<tr>
<td>Genre classification</td>
<td>novel</td>
<td>{detective, romance, gothic, … }</td>
</tr>
<tr>
<td>Sentiment classification</td>
<td>text</td>
<td>{positive, negative, neutral, mixed}</td>
</tr>
</tbody>
</table>
## Some Direct Text Classification Applications

<table>
<thead>
<tr>
<th>Task</th>
<th>$x$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language identification</td>
<td>text</td>
<td>{English, Mandarin, Greek, ...}</td>
</tr>
<tr>
<td>Spam classification</td>
<td>email</td>
<td>{spam, not spam}</td>
</tr>
<tr>
<td>Authorship attribution</td>
<td>text</td>
<td>{jk rowling, james joyce, ... }</td>
</tr>
<tr>
<td>Genre classification</td>
<td>novel</td>
<td>{detective, romance, gothic, ... }</td>
</tr>
<tr>
<td>Sentiment classification</td>
<td>text</td>
<td>{positive, negative, neutral, mixed}</td>
</tr>
</tbody>
</table>

Indirectly, methods from text classification apply to a huge range of settings in natural language processing, and will appear again and again throughout the course.
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 Bag-of-Words

- One challenge is that the sequential representation \((w_1, w_2, ..., w_T)\) may have a different length \(T\) for every document.

- The bag-of-words is a fixed-length representation, which consists of a vector of word counts:

  \[
  \textbf{w} = \text{It was the best of times, it was the worst of times}
  \]

  \[
  \textbf{x} = [\text{aardvark}, ..., \text{best}, ..., \text{it}, ..., \text{of}, ..., \text{zyther}]_{0,1,2,2,0}
  \]

- The length of \(\textbf{x}\) is equal to the size of the vocabulary \(V\)

- For each \(\textbf{x}\), there may be many possible \(\textbf{w}\), depending on word order.
Let $\psi(x, y)$ score the compatibility of bag-of-words $x$ and label $y$, then

$$\hat{y} = \arg\max_y \psi(x, y)$$

In a linear classifier, this scoring function has a simple form:

$$\psi(x, y) = \theta \cdot f(x, y) = \sum_{j=1}^{n} \theta_j \cdot f_j(x, y)$$

where $\theta$ is a vector of weights, and $f$ is a feature function.
Feature Functions

- In classification, the feature function is usually a simple combination of \( x \) and \( y \), such as:

\[
f_j(x, y) = \begin{cases} 
  x_{whale}, & \text{if } y = \text{FICTION} \\
  0, & \text{otherwise}
\end{cases}
\]
To summarize, our classification function is:

$$\hat{y} = \underset{y}{\text{argmax}}\ \theta \cdot f(x, y)$$

where $x$ is the bag-of-words representation, and $f$ is a feature function.

The learning problem is to find the right weights $\theta$, assuming a labeled dataset $\{(x^{(i)}, y^{(i)})\}_{i=1}^N$. \[\text{Your text here}\]
Naïve Bayes is a probabilistic classifier. It takes the following strategy:

- Define a probability model $p(x, y)$
- Estimate the parameters of the probability model by maximum likelihood – that is, by maximizing the likelihood of the dataset
First, assume each instance is independent of the others

\[ p(x^{(1:N)}, y^{(1:N)}) = \prod_{i=1}^{N} p(x^{(i)}, y^{(i)}) \]

Apply the chain rule of the probability

\[ p(x, y) = p(x|y) \cdot p(y) \]

Define the parametric form of each probability

\[ p(y) = \text{Categorical}(\mu) \quad p(x|y) = \text{Multinomial}(\phi) \]

The multinomial is a distribution over vectors of counts

The parameters \( \mu \) and \( \phi \) are vectors of probabilities
Suppose the word *whale* has probability $\phi_j$

- What is the probability that this word appears 3 times?
The Multinomial Distribution

Each word’s probability is exponentiated by its count,

\[ \text{Multinomial}(x, \phi, T) = \prod_{j=1}^{V} \phi_j^{x_j} \]
The Multinomial Distribution

Each word’s probability is exponentiated by its count,

- $\text{Multinomial}(x, \phi, T) = \prod_{j=1}^V \phi_j^{x_j}$
- The coefficient is the count of the number of possible orderings of $x$.

\[
\binom{n}{n_1, n_2, \ldots, n_k} = \frac{n!}{n_1! n_2! \cdots n_k!}
\]
The Multinomial Distribution

Each word’s probability is exponentiated by its count,

- Multinomial(\(x, \phi, T\)) = \(\frac{(\sum_{j=1}^{V} x_j)!}{\prod_{j=1}^{V} (x_j)!} \prod_{j=1}^{V} \phi_j^{x_j}\)

- The coefficient is the count of the number of possible orderings of \(x\).
- Crucially, it does not depend on the frequency parameter \(\phi\)
Estimating Naïve Bayes

- In relative frequency estimation, the parameters are set to empirical frequencies:

\[
\hat{\phi}_{y,j} = \frac{\text{count}(y, j)}{\sum_{j' = 1}^{V} \text{count}(y, j')} = \frac{\sum_{i:y^{(i)} = y} x_{j}^{(i)}}{\sum_{j' = 1}^{V} \sum_{i:y^{(i)} = y} x_{j'}^{(i)}}
\]

\[
\hat{\mu}_{y} = \frac{\text{count}(y)}{\sum_{y'} \text{count}(y')}. 
\]

- This turns out to be identical to the maximum likelihood estimate:

\[
\hat{\phi}, \hat{\mu} = \arg\max_{\phi, \mu} \prod_{i=1}^{N} p(x^{(i)}, y^{(i)}) = \arg\max_{\phi, \mu} \sum_{i=1}^{N} \log p(x^{(i)}, y^{(i)})
\]
Quick Question (1)

Multiplying lots of small probabilities (all are under 1) can lead to numerical underflow ...
Quick Question (1)

Multiplying lots of small probabilities (all are under 1) can lead to numerical underflow ...

$$\log \prod_{i} x_i = \sum_{i} \log x_i$$
Low Count Issue

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

\[
\hat{p}(\text{“fantastic” } | \text{positive}) = \frac{\text{count(“fantastic”, positive)}}{\sum_{w \in V} \text{count}(w, \text{positive})} = 0
\]

- Zero probabilities cannot be conditioned away
Smoothing

- To deal with low counts, it can be helpful to smooth probabilities

\[ \hat{\phi}_{y,j} = \frac{\alpha + \text{count}(y,j)}{V\alpha + \sum_{j'=1}^{V} \text{count}(y,j')} \]

- Smoothing term \( \alpha \) is a hyperparameter, which must be tuned on a development set
- Laplace (add-1) smoothing: widely used
Naïve Bayes is so called because:

- Bayes rule is used to convert the observation probability \( p(x|y) \) into the label probability \( p(y|x) \)
- The multinomial distribution naively ignores dependencies between words, and treats every word as equally informative
- Discriminative classifiers avoid this problem by not attempting to model the “generative” probability \( p(x) \)
The Perceptron Classifier

- Error-driven rather than independence assumption
The Perceptron Classifier

- A simple learning rule:
  - Run the current classifier on an instance in the training data, obtaining $\hat{y} = \arg\max_y \psi(x^{(i)}, y)$
  - If the prediction is incorrect:
    - Increase the weights for the features of the true label
    - Decrease the weights for the features of the predicted label
    - $\theta \leftarrow \theta + f(x^{(i)}, y^{(i)}) - f(x^{(i)}, \hat{y})$
  - Repeat until all training instances are correctly classified, or run out of time
The Perceptron Classifier (Online Learning)

Algorithm 3 Perceptron learning algorithm

1: procedure PERCEPTRON($x^{(1:N)}$, $y^{(1:N)}$)
2: $t \leftarrow 0$
3: $\theta^{(0)} \leftarrow 0$
4: repeat
5: $t \leftarrow t + 1$
6: Select an instance $i$
7: $\hat{y} \leftarrow \arg\max_y \theta^{(t-1)} \cdot f(x^{(i)}, y)$
8: if $\hat{y} \neq y^{(i)}$ then
9: $\theta^{(t)} \leftarrow \theta^{(t-1)} + f(x^{(i)}, y^{(i)}) - f(x^{(i)}, \hat{y})$
10: else
11: $\theta^{(t)} \leftarrow \theta^{(t-1)}$
12: until tired
13: return $\theta^{(t)}$
Loss Function

- Many classifiers can be viewed as minimizing a loss function on the weights.
- Such a function should have two properties:
  - It should be a good proxy for the accuracy of the classifier
  - It should be easy to optimize
This perceptron can be viewed as optimizing the loss function

$$\ell_{\text{PERCEPTRON}}(\theta; x^{(i)}, y^{(i)}) = \max_{y \in Y} \theta \cdot f(x^{(i)}, y) - \theta \cdot f(x^{(i)}, y^{(i)}).$$
Perceptron as Gradient Descent

- This perceptron can be viewed as optimizing the loss function

\[
\ell_{\text{PERCEPTRON}}(\theta; x^{(i)}, y^{(i)}) = \max_{y \in Y} \theta \cdot f(x^{(i)}, y) - \theta \cdot f(x^{(i)}, y^{(i)}).
\]

- The gradient of the perceptron loss is part of the perceptron update

\[
\frac{\partial}{\partial \theta} \ell_{\text{perceptron}} = -f(x^{(i)}, y^{(i)}) + f(x^{(i)}, \hat{y})
\]

\[
\theta^{(t+1)} \leftarrow \theta^{(t)} - \frac{\partial}{\partial \theta} \ell_{\text{perceptron}} \quad \text{(gradient descent!)}
\]

\[
= \theta^{(t)} + f(x^{(i)}, y^{(i)}) - f(x^{(i)}, \hat{y}).
\]
Logistic Regression

- Perceptron classification is discriminative – learns to discriminate correct and incorrect labels
- Naïve Bayes is probabilistic: it assigns calibrated confidence scores to its predictions
- Logistic regression is both discriminative and probabilistic. It directly computes the conditional probability of the label:

\[
p(y \mid x; \theta) = \frac{\exp(\theta \cdot f(x, y))}{\sum_{y' \in Y} \exp(\theta \cdot f(x, y'))}
\]
Logistic regression is both discriminative and probabilistic. It directly computes the conditional probability of the label:

\[ p(y \mid x; \theta) = \frac{\exp(\theta \cdot f(x, y))}{\sum_{y' \in Y} \exp(\theta \cdot f(x, y'))} \]

- Exponentiation ensures that the probabilities are non-negative.
Logistic regression is both discriminative and probabilistic. It directly computes the conditional probability of the label:

\[
p(y \mid x; \theta) = \frac{\exp(\theta \cdot f(x, y))}{\sum_{y' \in \mathcal{Y}} \exp(\theta \cdot f(x, y'))}
\]

- Exponentiation ensures that the probabilities are non-negative.
- Normalization ensures that the probabilities sum to one.
Learning Logistic Regression

- **Maximization** of the conditional log-likelihood

\[
\log p(y^{(1:N)} | x^{(1:N)}; \theta) = \sum_{i=1}^{N} \log p(y^{(i)} | x^{(i)}; \theta) = \sum_{i=1}^{N} \theta \cdot f(x^{(i)}, y^{(i)}) - \log \sum_{y' \in \mathcal{Y}} \exp \left( \theta \cdot f(x^{(i)}, y') \right)
\]
Learning Logistic Regression

- **Maximization** of the conditional log-likelihood
- **Minimization** of the negative log-likelihood/logistic loss

\[
\ell_{\text{LOGREG}}(\theta; x^{(i)}, y^{(i)}) = -\theta \cdot f(x^{(i)}, y^{(i)}) + \log \sum_{y' \in \mathcal{Y}} \exp(\theta \cdot f(x^{(i)}, y'))
\]
Regularization

- Learning can often be made more robust by regularization: penalizing large weights

\[ \min_{\theta} \sum_{i=1}^{N} \ell_{\text{LOGREG}}(\theta; x^{(i)}, y^{(i)}) + \lambda \|\theta\|_2^2 \]

- where the scalar \( \lambda \) controls the strength of regularization, and \( \|\theta\|_2^2 = \sum_j \theta_j^2 \).
Logistic regression, perceptron both learn by minimizing a loss function. A general strategy for minimization is gradient descent

\[ \theta^{(t+1)} \leftarrow \theta^{(t)} - \eta^{(t)} \nabla_{\theta} L \]

where \( \eta^{(t)} \in \mathbb{R}_+ \) is the learning rate at iteration \( t \)
Stochastic Gradient Descent (Online Optimization)

- Computing the gradient over all instances is expensive
- **Stochastic gradient descent** approximates the gradient by its value on a single data:

\[
\sum_{i=1}^{N} \ell(\theta; x^{(i)}, y^{(i)}) \approx N \times \ell(\theta; x^{(j)}, y^{(j)})
\]

- \((x^{(i)}, y^{(i)})\) is sampled at random from the training set \((x^{(j)}, y^{(j)}) \sim \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}\)

Theoretically guaranteed!
Online Optimization

- **Gradient descent** computes the gradient over all instances.
- **Stochastic gradient descent** approximates the gradient by its value on a single data point.
- **Minibatch** gradient descent approximates the gradient by its value on a small number of instances. This is suited to GPU architectures, widely used in deep learning.
Generalized Gradient Descent

Algorithm 5 Generalized gradient descent. The function \textsc{Batcher} partitions the training set into $B$ batches such that each instance appears in exactly one batch. In gradient descent, $B = 1$; in stochastic gradient descent, $B = N$; in minibatch stochastic gradient descent, $1 < B < N$.

\begin{algorithm}
\begin{algorithmic}[1]
\Procedure{Gradient-Descent}{$\mathbf{x}^{(1:N)}, \mathbf{y}^{(1:N)}, L, \eta^{(1\ldots\infty)}, \textsc{Batcher}, T_{\text{max}}$}
\State $\theta \leftarrow 0$
\State $t \leftarrow 0$
\Repeat
\State $(b^{(1)}, b^{(2)}, \ldots, b^{(B)}) \leftarrow \textsc{Batcher}(N)$
\For{$n \in \{1, 2, \ldots, B\}$}
\State $t \leftarrow t + 1$
\State $\theta(t) \leftarrow \theta(t-1) - \eta(t) \nabla_{\theta} L(\theta(t-1); \mathbf{x}^{(b^{(1)}_n), b^{(2)}_n, \ldots}, \mathbf{y}^{(b^{(1)}_n), b^{(2)}_n, \ldots})$
\If{Converged(\theta^{(1,2,\ldots,t)})}
\State \Return $\theta(t)$
\EndIf
\EndFor
\Until{$t \geq T_{\text{max}}$}
\State \Return $\theta(t)$
\EndProcedure
\end{algorithmic}
\end{algorithm}
### Summary of Linear Classification

<table>
<thead>
<tr>
<th></th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Bayes</td>
<td>Simple, probabilistic, fast</td>
<td>Not very accurate</td>
</tr>
<tr>
<td></td>
<td>Closed-form solution</td>
<td></td>
</tr>
<tr>
<td>Perceptron</td>
<td>Simple, accurate</td>
<td>Not probabilistic, may overfit</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>Error-driven learning, regularized</td>
<td>More difficult to implement</td>
</tr>
</tbody>
</table>