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 15th, 3:00pm, EST

- Other Questions?

Very Quick Review on Probabilities

- Event space (e.g., $\mathcal{X}, \mathcal{Y}$) – in this class, usually discrete
- Random variables (e.g., X, Y)
- Random variable X takes value $x, x \in \mathcal{X}$ with probability $p(X = x)$ or $p(x)$

Very Quick Review on Probabilities

- Joint probability $p(X = x, Y = y)$
- Conditional probability $p(X = x \mid Y = y) = \frac{p(X=x,Y=y)}{p(Y=y)}$

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:
  - $p(X = x, Y = y) = p(X = x) \cdot p(Y = y)$

Very Quick Review on Probabilities

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

- The number of ways to select k words out of n given words (“unordered samples without replacement”)

$$\binom{n}{n_1, n_2, \dots, n_k} = \frac{n!}{n_1!\, n_2! \cdots n_k!}$$

- Here, $n, n_1, n_2 \dots, n_k$ 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, . . . , n_k, respectively

Classification

- A mapping h from input data x $(\text{drawn from instance space } \mathcal{X})$ 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

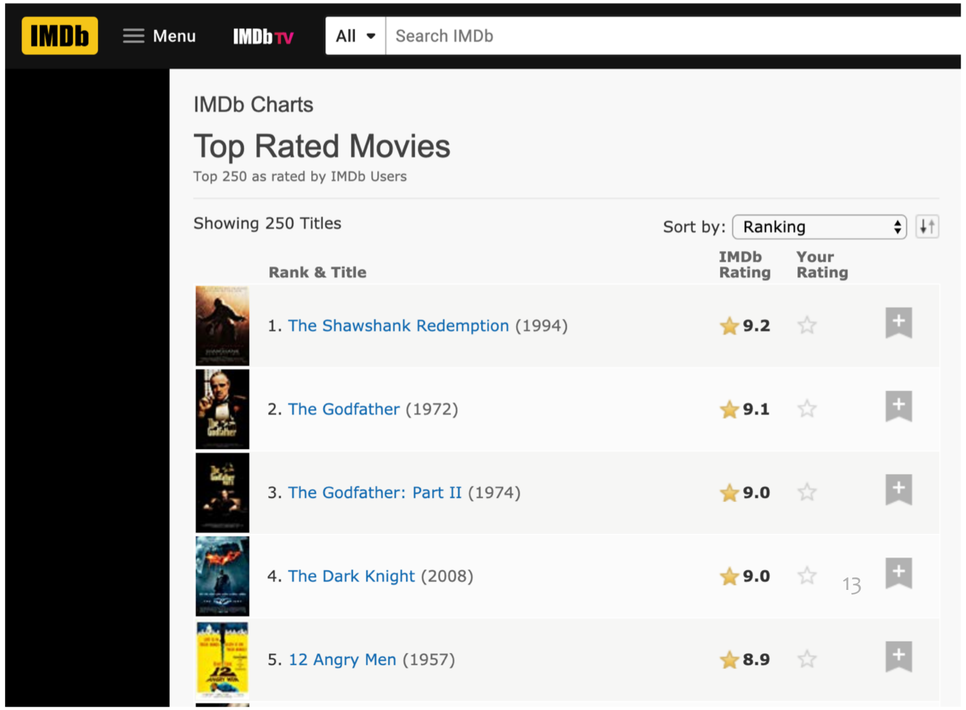
"… is a film which still causes real, not figurative, chills to run along my spine, and it is certainly the bravest and most ambitious fruit of Coppola's genius"

Roger Ebert, Apocalypse Now

- "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

negative



IMDb Charts

Top Rated Movies

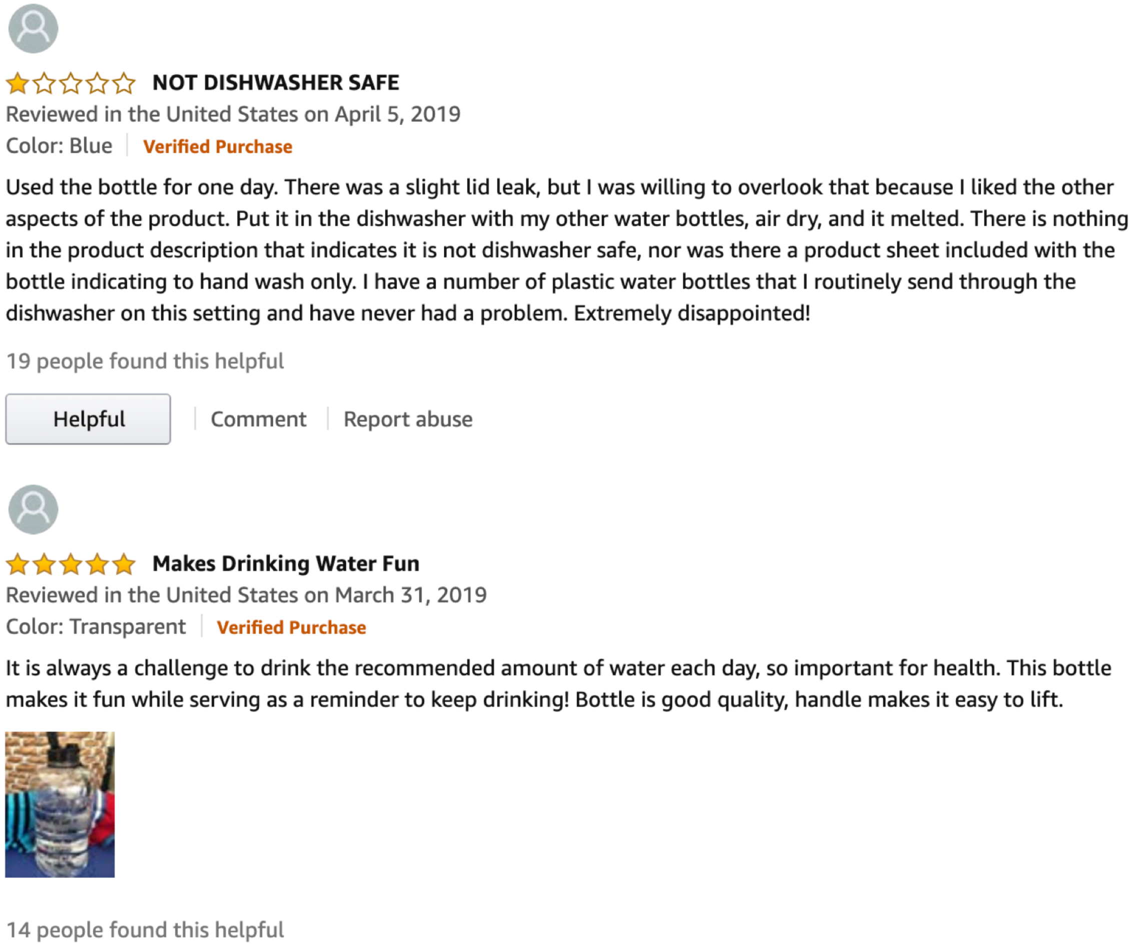
Top 250 as rated by IMDb Users

Showing 250 Titles

Sort by: Ranking

Rank & Title	IMDb Rating	Your Rating
1. The Shawshank Redemption (1994)	★9.2	☆
2. The Godfather (1972)	★9.1	☆
3. The Godfather: Part II (1974)	★9.0	☆
4. The Dark Knight (2008)	★9.0	☆
5. 12 Angry Men (1957)	★8.9	☆

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

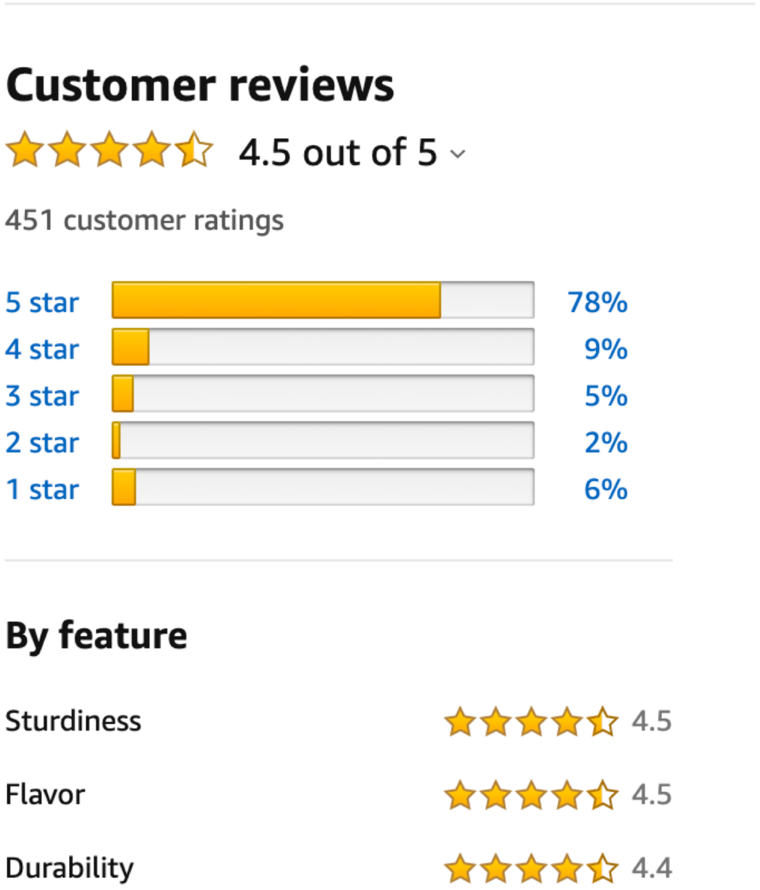
★★★★★ **Makes Drinking Water Fun**

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

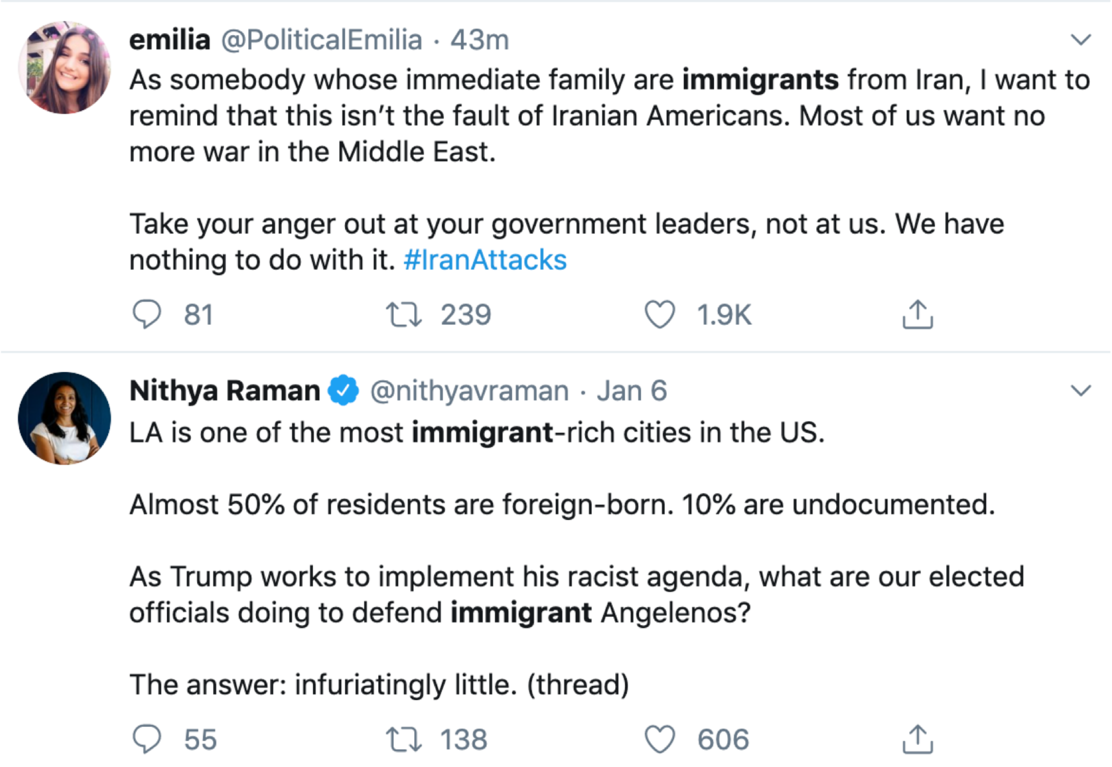
451 customer ratings

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 @nithyavraman · Jan 6

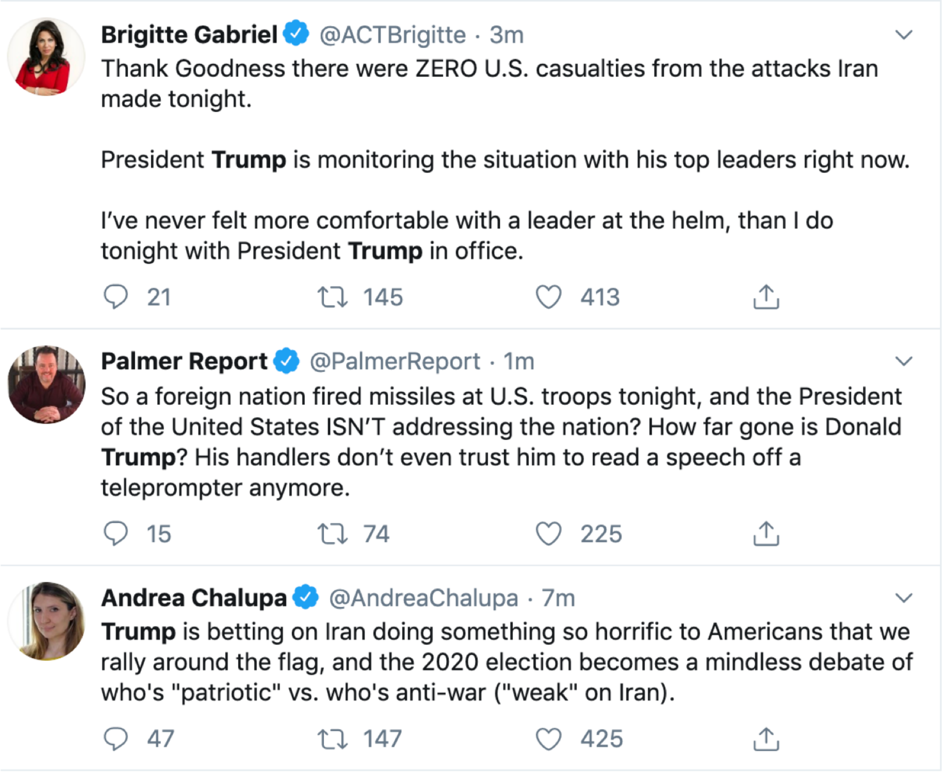
LA is one of the most **immigrant**-rich cities in the US.

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?

The answer: infuriatingly little. (thread)

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).

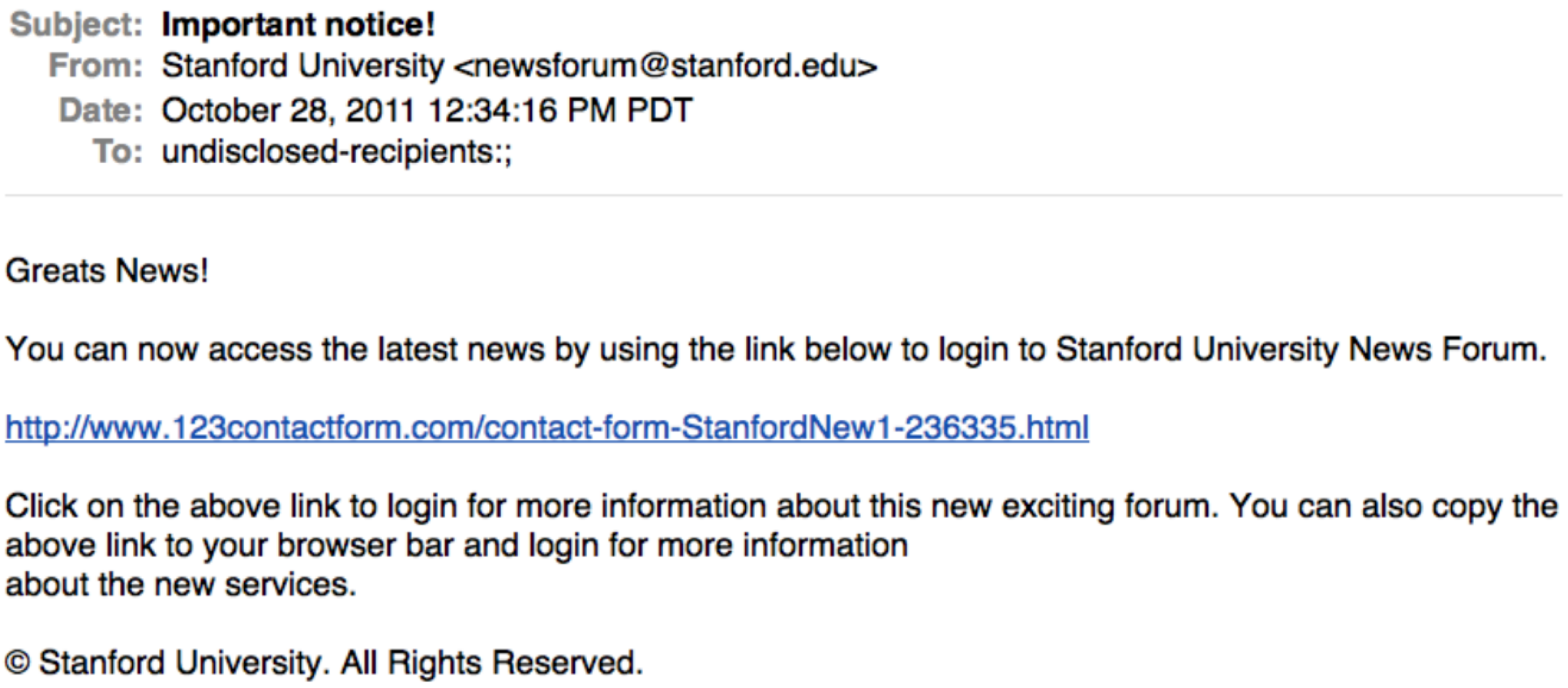
47 147 425

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…

S. Argamon, M. Koppel, J. Fine, A. R. Shimoni, 2003. “Gender, Genre, and Writing Style in Formal Written Texts,” Text, volume 23, number 3, pp. 321–346

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.

http://www.123contactform.com/contact-form-StanfordNew1-236335.html

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

Brain and Cognition

Syntactic frame and verb bias in aphasia: Plausibility judgments of undergoer-subject sentences

Susanne Gahl, Lise Menn, Gail Ramsberger, Daniel S. Jurafsky, Elizabeth Elder, Molly Rewega, and L. Holland Audrey

?

MeSH Subject Category Hierarchy

- Antogonists 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 $\boldsymbol{w} = (w_1, w_2, \dots, w_T) \in \mathcal{V}^*$, predict a label $y \in \mathcal{Y}$

Some Direct Text Classification Applications

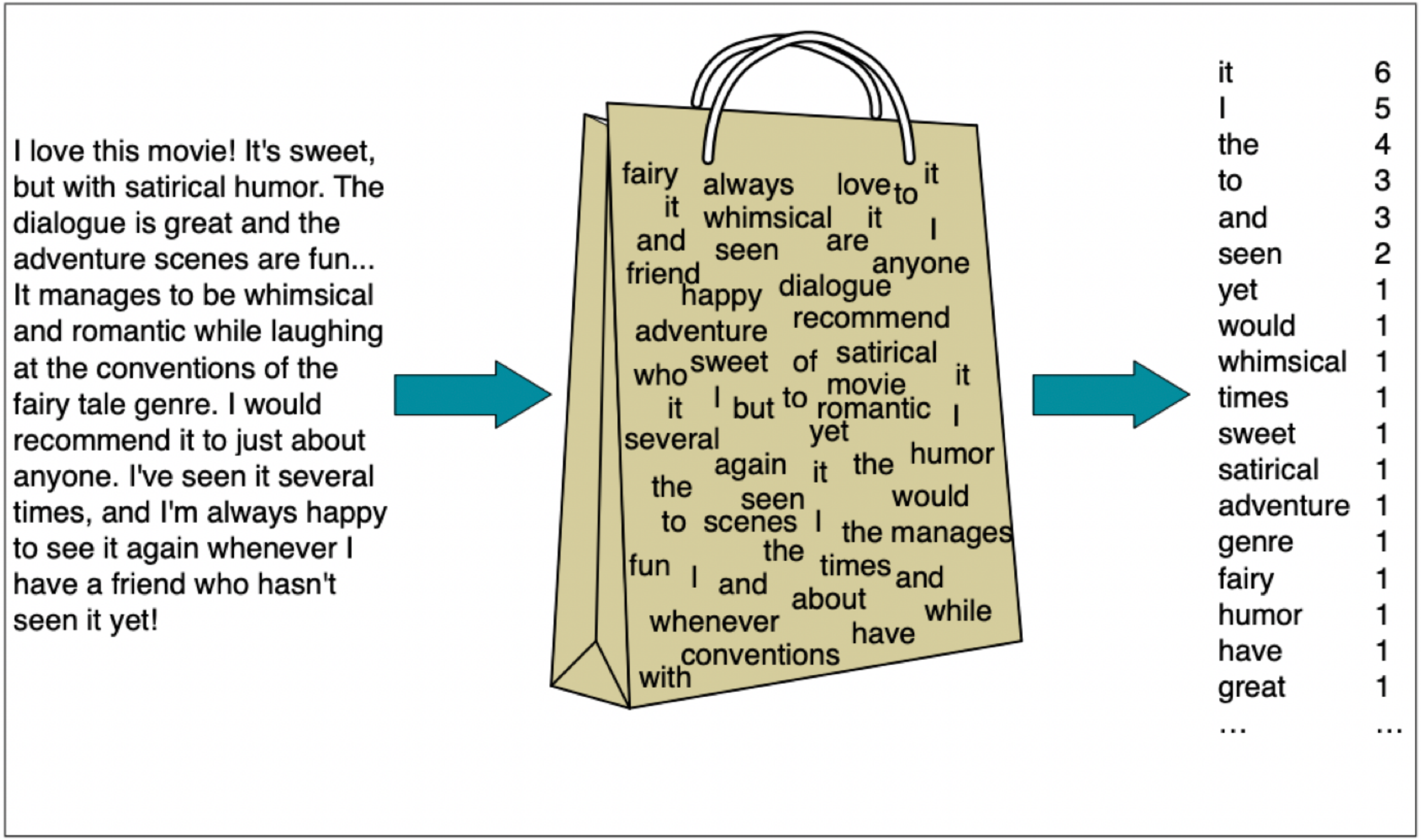
Task	$\boldsymbol{x}$	$\boldsymbol{y}$
Language identification	text	{English, Mandarin, Greek, … }
Spam classification	email	{spam, not spam}
Authorship attribution	text	{jk rowling, james joyce, … }
Genre classification	novel	{detective, romance, gothic, … }
Sentiment classification	text	{positive, negative, neutral, mixed}

Some Direct Text Classification Applications

Task	x	y
Language identification	text	{English, Mandarin, Greek, … }
Spam classification	email	{spam, not spam}
Authorship attribution	text	{jk rowling, james joyce, … }
Genre classification	novel	{detective, romance, gothic, … }
Sentiment classification	text	{positive, negative, neutral, mixed}

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.

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!

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
…	…

The Bag-of-Words

- One challenge is that the sequential representation $(w_1, w_2, \dots, w_T)$ 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:

$$\boldsymbol{w} = \text{It was the best of times, it was the worst of times}$$

$$\boldsymbol{x} = [\underbrace{\text{aardvark}}_{0}, \dots, \underbrace{\text{best}}_{1}, \dots, \underbrace{\text{it}}_{2}, \dots, \underbrace{\text{of}}_{2}, \dots, \underbrace{\text{zyther}}_{0}]$$

  - The length of $\boldsymbol{x}$ is equal to the size of the vocabulary V
- For each $\boldsymbol{x}$, there may be many possible **w**, depending on word order.

Linear Classification on the Bag of Words

- Let $\psi(\boldsymbol{x}, y)$ score the compatibility of bag-of-words $\boldsymbol{x}$ and label y, then

$$\hat{y} = \underset{y}{\operatorname{argmax}}\, \psi(\boldsymbol{x}, y)$$

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

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

  - where $\boldsymbol{\theta}$ is a vector of weights, and $\boldsymbol{f}$ is a feature function

Feature Functions

- In classification, the feature function is usually a simple combination of $\boldsymbol{x}$ and y, such as:

$$f_j(\boldsymbol{x}, y) = \begin{cases} x_{whale}, & \text{if y} = \text{FICTION} \\ 0, & \text{otherwise} \end{cases}$$

Summary and Next Steps

- To summarize, our classification function is:

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

  where $\boldsymbol{x}$ is the bag-of-words representation, and $\boldsymbol{f}$ is a feature function

- The learning problem is to find the right weights $\boldsymbol{\theta}$, assuming a labeled dataset $\left\{\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N}$

Probabilistic Classification

- Naïve Bayes is a probabilistic classifier. It takes the following strategy:
  - Define a probability model $p(\boldsymbol{x}, y)$
  - Estimate the parameters of the probability model by maximum likelihood – that is, by maximizing the likelihood of the dataset

A Probability Model for Text Classification

- First, assume each instance is independent of the others
  - $p\left(\boldsymbol{x}^{(1:N)}, y^{(1:N)}\right) = \prod_{i=1}^{N} p(\boldsymbol{x}^{(i)}, y^{(i)})$
- Apply the chain rule of the probability
  - $p(\boldsymbol{x}, y) = p(\boldsymbol{x}|y) \cdot p(y)$
- Define the parametric form of each probability
  - $p(y) = \text{Categorical}(\mu) \qquad p(\boldsymbol{x}|y) = \text{Multinomail}(\phi)$
  - The multinomial is a distribution over vectors of counts
  - The parameters μ and ϕ are vectors of probabilities

The Multinomial Distribution

- Suppose the word *whale* has probability ϕ_j
  - What is the probability that this word appears 3 times?

The Multinomial Distribution

Each word’s probability is exponentiated by its count,

- $\text{Multinomail}(\boldsymbol{x}, \phi, T) = \qquad \prod_{j=1}^{V} \phi_j^{x_j}$

The Multinomial Distribution

Each word’s probability is exponentiated by its count,

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

$$\boxed{\binom{n}{n_1, n_2, \dots, n_k} = \frac{n!}{n_1!\, n_2! \cdots n_k!}}$$

The Multinomial Distribution

Each word’s probability is exponentiated by its count,

- $\text{Multinomail}(\boldsymbol{x}, \phi, T) = \frac{(\sum_{j=1}^{V} x_j)!}{\prod_{j=1}^{V}(x_j!)} \prod_{j=1}^{V} \phi_j^{x_j}$
- The coefficient is the count of the number of possible orderings of $\boldsymbol{x}$.
- Crucially, it does not depend on the frequency parameter ϕ

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:

$$\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\mu}} = \underset{\boldsymbol{\phi},\boldsymbol{\mu}}{\text{argmax}} \prod_{i=1}^{N} \mathrm{p}(\boldsymbol{x}^{(i)}, y^{(i)}) = \underset{\boldsymbol{\phi},\boldsymbol{\mu}}{\text{argmax}} \sum_{i=1}^{N} \log \mathrm{p}(\boldsymbol{x}^{(i)}, y^{(i)})$$

Quick Question (1)

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

$$\prod_i x_i$$

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* ?

- $\hat{p}(\text{“}fantastic\text{”}\,|positive) = \frac{count(\text{“}fantastic\text{”}, positive)}{\sum_{w \in V} count(w, positive)}$ = 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 α is a hyperparameter, which must be tuned on a development set
- Laplace (add-1)smoothing: widely used

Too Naïve?

- Naïve Bayes is so called because:
  - Bayes rule is used to convert the observation probability $p(\boldsymbol{x}|y)$ into the label probability $p(\boldsymbol{y}|\boldsymbol{x})$
  - 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 $p(\boldsymbol{x})$

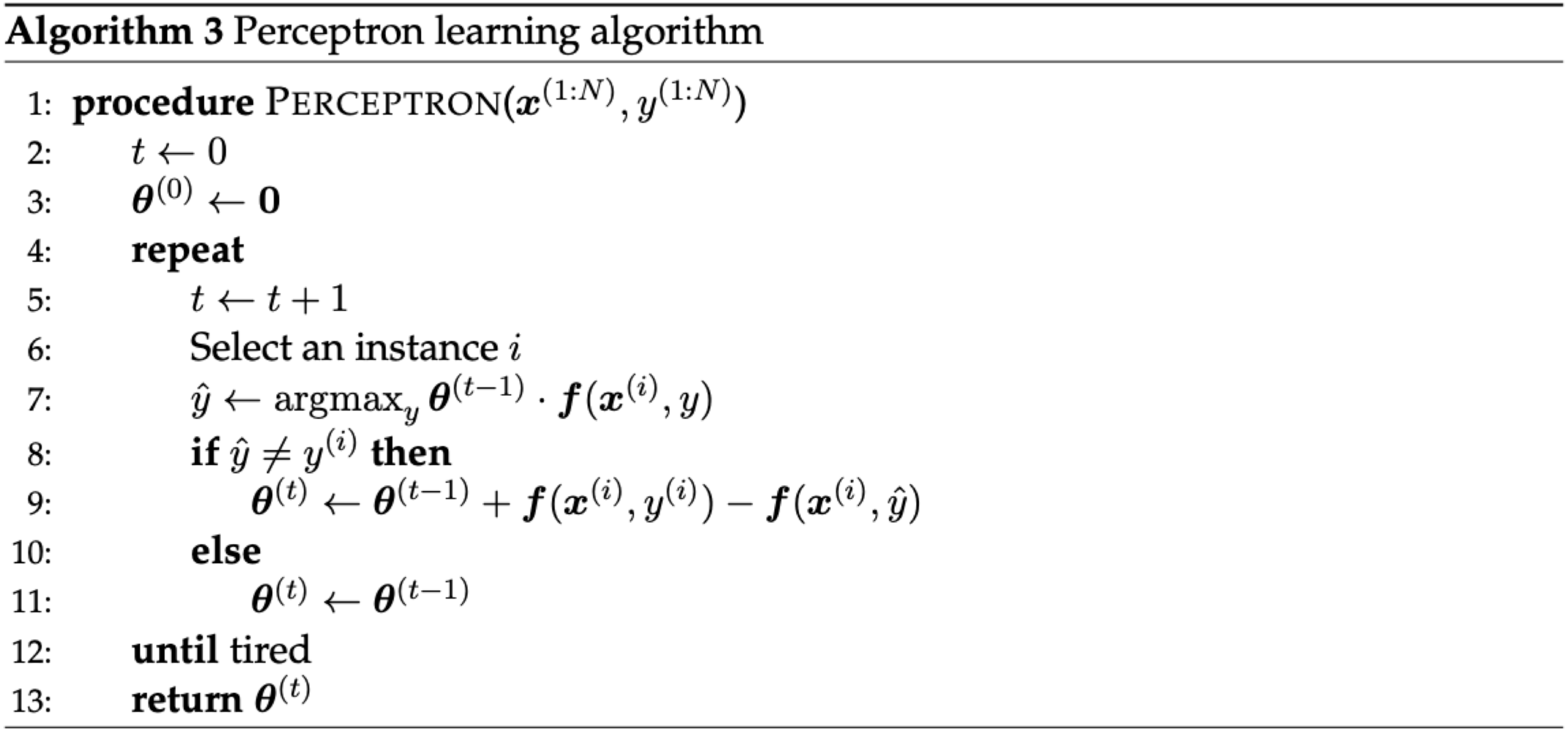
The Perceptron Classifier

- 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 $\hat{y} = \operatorname*{argmax}_{y} \psi(\boldsymbol{x}^{(i)}, y)$
  - 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
      - $\theta \leftarrow \theta + \boldsymbol{f}\left(\boldsymbol{x}^{(\boldsymbol{i})}, y^{(i)}\right) - \boldsymbol{f}\left(\boldsymbol{x}^{(\boldsymbol{i})}, \hat{y}\right)$
  - Repeat until all training instances are correctly classified, or run out of time

The Perceptron Classifier (Online Learning)



Algorithm 3 Perceptron learning algorithm

```
procedure PERCEPTRON($\boldsymbol{x}^{(1:N)}, y^{(1:N)}$)
    $t \leftarrow 0$
    $\boldsymbol{\theta}^{(0)} \leftarrow \mathbf{0}$
    repeat
        $t \leftarrow t + 1$
        Select an instance $i$
        $\hat{y} \leftarrow \text{argmax}_y \, \boldsymbol{\theta}^{(t-1)} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y)$
        if $\hat{y} \neq y^{(i)}$ then
            $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)} + \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) - \boldsymbol{f}(\boldsymbol{x}^{(i)}, \hat{y})$
        else
            $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)}$
    until tired
    return $\boldsymbol{\theta}^{(t)}$
```

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

Perceptron as Gradient Descent

- This perceptron can be viewed as optimizing the loss function

$$\ell_{\textsc{Perceptron}}(\boldsymbol{\theta}; \boldsymbol{x}^{(i)}, y^{(i)}) = \max_{y \in \mathcal{Y}} \boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y) - \boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)})$$

Perceptron as Gradient Descent

- This perceptron can be viewed as optimizing the loss function

$$\ell_{\text{PERCEPTRON}}(\boldsymbol{\theta}; \boldsymbol{x}^{(i)}, y^{(i)}) = \max_{y \in \mathcal{Y}} \boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y) - \boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}),$$

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

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

$$\begin{aligned} \boldsymbol{\theta}^{(t+1)} \leftarrow & \boldsymbol{\theta}^{(t)} - \frac{\partial}{\partial \boldsymbol{\theta}} \ell_{\text{perceptron}} & \text{(gradient descent!)} \\ = & \boldsymbol{\theta}^{(t)} + \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) - \boldsymbol{f}(\boldsymbol{x}^{(i)}, \hat{y}). \end{aligned}$$

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:

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

Logistic Regression

- Logistic regression is both discriminative and probabilistic. It directly computes the conditional probability of the label:

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

  - Exponentiation ensures that the probabilities are non-negative.

Logistic Regression

- Logistic regression is both discriminative and probabilistic. It directly computes the conditional probability of the label:

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

  - 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

$$
\begin{aligned}
\log \mathrm{p}(\boldsymbol{y}^{(1:N)} \mid \boldsymbol{x}^{(1:N)}; \boldsymbol{\theta}) &= \sum_{i=1}^{N} \log \mathrm{p}(y^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \\
&= \sum_{i=1}^{N} \boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) - \log \sum_{y' \in \mathcal{Y}} \exp\left(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')\right)
\end{aligned}
$$

Learning Logistic Regression

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

$$\ell_{\textsc{LogReg}}(\boldsymbol{\theta}; \boldsymbol{x}^{(i)}, y^{(i)}) = -\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \log \sum_{y' \in \mathcal{Y}} \exp(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y'))$$

Regularization

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

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \ell_{\mathrm{LogReg}}(\boldsymbol{\theta}; \boldsymbol{x}^{(i)}, y^{(i)}) + \lambda ||\boldsymbol{\theta}||_2^2$$

- where the scalar λ controls the strength of regularization, and $||\boldsymbol{\theta}||_2^2 = \sum_j \theta_j^2$

Gradient Descent (Batch Optimization)

- Logistic regression, perceptron both learn by minimizing a loss function. A general strategy for minimization is gradient descent

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \eta^{(t)} \nabla_{\boldsymbol{\theta}} L$$

- 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:

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

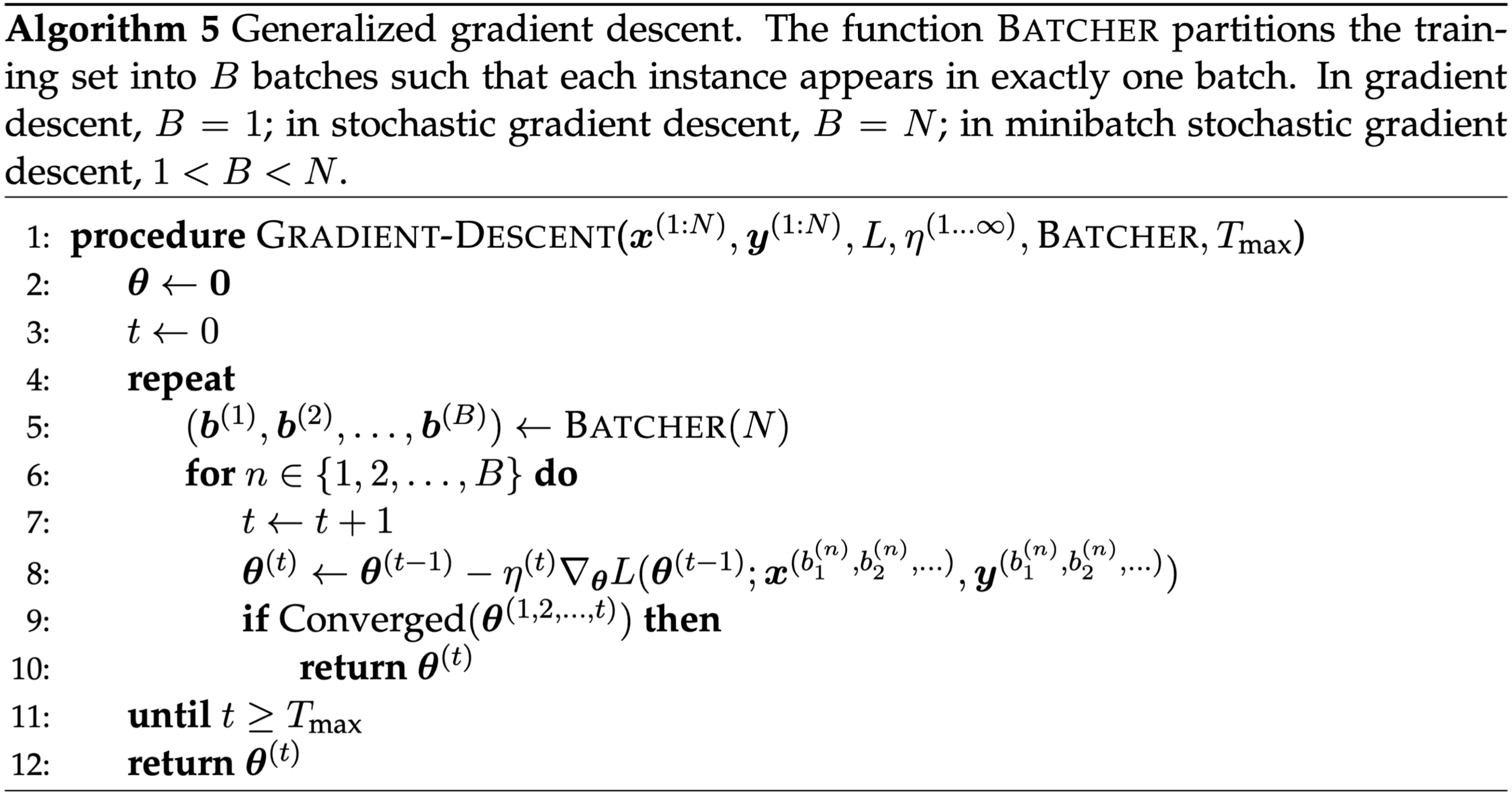
theoretically guaranteed!

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

Online Optimization

- Gradient descent computes the gradient over all instances
- Stochastic gradient descent approximates the gradient by its value on a single data
- Minibatch gradient descent approximates the gradient by its value on 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 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, $1 < B < N$.

```
procedure GRADIENT-DESCENT($\boldsymbol{x}^{(1:N)}, \boldsymbol{y}^{(1:N)}, L, \eta^{(1\dots\infty)}$, BATCHER, $T_{\text{max}}$)
    $\boldsymbol{\theta} \leftarrow \mathbf{0}$
    $t \leftarrow 0$
    repeat
        $(\boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}, \dots, \boldsymbol{b}^{(B)}) \leftarrow$ BATCHER$(N)$
        for $n \in \{1, 2, \dots, B\}$ do
            $t \leftarrow t + 1$
            $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)} - \eta^{(t)} \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}^{(t-1)}; \boldsymbol{x}^{(b_1^{(n)}, b_2^{(n)}, \dots)}, \boldsymbol{y}^{(b_1^{(n)}, b_2^{(n)}, \dots)})$
            if Converged$(\boldsymbol{\theta}^{(1,2,\dots,t)})$ then
                return $\boldsymbol{\theta}^{(t)}$
    until $t \geq T_{\text{max}}$
    return $\boldsymbol{\theta}^{(t)}$
```

Summary of Linear Classification

	Pros	**Cons**
Naive Bayes	Simple, probabilistic, fast Closed-form solution	Not very accurate
Perceptron	Simple, accurate	Not probabilistic, may overfit
Logistic Regression	Error-driven learning, regularized	More difficult to implement