CSED 503: NLP and LMs

Text classification

Luke Zettlemoyer

lsz@cs.washington.edu

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

Is this spam?

from:	**ECRES 2022 <2022@ecres.net>** via amazonses.com
reply-to:	2022@ecres.net
to:	yuliats@cs.washington.edu
date:	Feb 22, 2022, 7:21 AM
subject:	The Best Renewable Energy Conference (Last chance !)
signed-by:	amazonses.com
security:	Standard encryption (TLS) Learn more

Dear Colleague,

Account: yuliats@cs.washington.edu

Good news: Due to many requests, the submission deadline has been extended to 10 March 2022 (It is firm date).

We would like to invite you to submit a paper to 10. European Conference on Renewable Energy Systems (ECRES). **ECRES 2022 will be held hybrid mode, the participants can present their papers physically or online.** The event is going to be organized in Istanbul/Turkey under the technical sponsorship of Istanbul Medeniyet University and many international institutions. The conference is highly international with the participants from all continents and more than 40 countries.

The submission deadline and special and regular issue journals can be seen in ecres.net

There will be keynote speakers who will address specific topics of energy as you would see at ecres.net/keynotes.html

CLICK FOR PAPER SUBMISSION

All accepted papers will be published in a special Conference Proceedings under a specific ISBN. Besides, the extended versions will be delivered to reputable journals **indexed in SCI, E-SCI, SCOPUS, and EBSCO.** You can check our previous journal publications from ecres.net . **Please note that the official journal of the event, Journal of Energy Systems (dergipark.org.tr/jes) is also indexed in SCOPUS.**

Spam classification

Dear Colleague,

Account: yuliats@cs.washington.edu

Good news: Due to many requests, the submission deadline has been extended to 10 March 2022 (It is firm date).

We would like to invite you to submit a pap nference on Renewable Energy Systems (ECRES). ECRES 20 , the participants can present their papers physically o to be organized in Istanbul/Turkey under the tech Medeniyet University and many international institutions. The rnational with the participants from all continents and more than 40

The submission deadline and spec and regular issue journals can be seen in ecres.net

There will be keynote speakers who will address specific topics of energy as you would see at ecres.net/keynotes.html

CLICK FOR PAPER SUBMISSION

All accepted papers will be published in a special Conference Proceedings under a specific ISBN. Besides, the extended versions will be delivered to reputable journals **indexed in SCI, E-SCI, SCOPUS, and EBSCO.** You can check our previous journal publications from ecres.net . **Please note that the official journal of the event Journal of Energy Systems (dergipark.org.tr/jes) is also indexed in SCOPUS.**

spam

Invitation to present at the February 2022 Wikimedia Research Showcase

Emily Lescak

Hi Yulia,

My name is Emily Lescak and I am a member of the Research team at the Wikimedia Foundation. On be present your research on social biases on Wikipedia at our Research Showcase in February 2022. This topic fits into our theme for this showcase, which is gaps and biases on Wikipedia.

The Wikimedia Research Showcase is a monthly, public lecture series where Foundation, ac work related to Wikipedia, Wikimedia, peer production, and open-source software. We focus on topics and projects that we think our audience— a global community of academic researchers, Wik staff—would find interesting and relevant to their work.

Research Showcase presentations are generally 20 minutes long, with an additio ite two presenters to every showcase. Most presenters choose to use slides to present their work.

The February showcase takes place on the 16th at 9:15AM Pacific / 17:15 UT uTube and also archived for later viewing on the Wikimedia Foundation's YouTube channel

If this date does not work for you, but you are still interested in giving a showcase we can discuss other options.

I hope to get a chance to see your work presented at the Research Showcase!

Sincerely,

Emily

not spam

Language ID

Аяны замд түр зогсон тэнгэрийн байдлыг ажиглаад хөдлөх зуур гутал дор шинэхэн орсон цас шаржигнан дуугарч байв. Цасны тухай бодол сонин юм. Хот хүрээ тийш цас орвол орно л биз гэсэн хэнэггүй бодол маань хөдөө талд, говийн ээрэм хөндийд, малын бэлчээрт, малчдын хотонд болохоор солигдож эргэцүүлэн бодох нь хачин. Цас хэр орсон бол?

Београд, 16. јун 2013. године – Председник Владе Републике Србије Ивица Дачић честитао је кајакашици златне медаље у олимпијској дисциплини К-1, 500 метара, као и у двоструко дужој стази освојене на првенству Европе у Португалији.

Beograd, 16. jun 2013. godine – Predsednik Vlade Republike Srbije Ivica Dačić čestitao je kajakašici zlatne medalje u olimpijskoj disciplini K-1, 500 metara, kao i u dvostruko dužoj stazi osvojene na prvenstvu Evrope u Portugaliji.

Nestrankarski Urad za vladno odgovornost ZDA je objavil eksplozivno mnenje, da je vlada predsednika Donalda Trumpa kršila zvezno zakonodajo, ko je zadrževala izplačilo kongresno potrjene vojaške pomoči Ukrajini zaradi političnih razlogov. Predstavniški dom kongresa je prav zaradi tega sprožil ustavno obtožbo proti Trumpu.

Language ID

Аяны замд түр зогсон тэнгэрийн байдлыг ажиглаад хөдлөх зуур гутал дор шинэхэн орсон цас шаржигнан дуугарч байв. Цасны тухай бодол сонин юм. Хот рвол орно л биз гэсэн хэнэггүй бодол маань хөдөө тал өндийд, малын бэлчээрт, малчдын хотонд болохоор солигдож эргэцүүлэн бодох нь хачин. Цас хэр орсон бол?

mongolian

Београд, 16. јун 2013. године – Председник Владе Републике Ср нештитао је кајакашици златне медаље у оли ини К-1, 500 метара, као и у двоструко дужој стази освојене на првенству Европе у Португалији.

serbian

Beograd, 16. jun 2013. godine – Predsednik Vlade Republike S titao je kajakašici zlatne medalje u o K-1, 500 metara, kao i u dvostruko dužoj stazi osvojene na prvenstvu Evrope u Portugaliji.

serbian

Nestrankarski Urad za vladno odgovornost ZDA je objavil eksplozivno mnenje, da je vlada predsednika Donalda Trumpa kršila zvezno zakonodajo, ko je zadrževala izplačilo k vojaške pomoči Ukrajini zaradi političnih razlogov. Predstavniški d v zaradi tega sprožil ustavno obtožbo proti Trumpu.

slovenian

Sentiment analysis

By **John Neal**

This review is from: **Accoutrements Horse Head Mask (Toy)**

When I turned State's Witness, they didn't have enough money to put me in the Witness Protection Program, so they bought me this mask and gave me a list of suggested places to move. Since then I've lived my life in peace and safety knowing that my old identity is forever obscured by this life-saving item.

By **Christine E. Torok**

Verified Purchase (What's this?)

First of all, for taste I would rate these a 5. So good. Soft, true-to-taste fruit flavors like the sugar variety...I was a happy camper.

BUT (or should I say BUTT), not long after eating about 20 of these all hell broke loose. I had a gastrointestinal experience like nothing I've ever imagined. Cramps, sweating, bloating beyond my worst nightmare. I've had food poisoning from some bad shellfish and that was almost like a skip in the park compared to what was going on inside

Sentiment analysis

By John Neal

This review is from: Accoutrements Horse Head Mask (Toy)

When I turned State's Witness, they didn't have enough money to put me in the Witness Protection Program, so they bought me this mask and some other suggested places to move. Since then I've lived my life in peace and safety knowing that my old identity is forever obscured by this life-saving item.

By Christine E. Torok

Verified Purchase (What's this?)

First of all, for taste I would rate these a 5. So good. Soft, true-to-taste fruit flavors like the sugar variety...I was a happy camper.

BUT (or should I say BUTT), not long after eating about 20 of these all hell broke loose. I had a gastrointestinal experience like nothing I've ever imagined. Cramps, sweating, bloating beyond my worst nightmare. I've had food poisoning from some bad shellfish and that was almost like a skip in the park compared to what was going on inside

Topic classification

MEDLINE Article

?

MeSH Subject Category Hierarchy

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

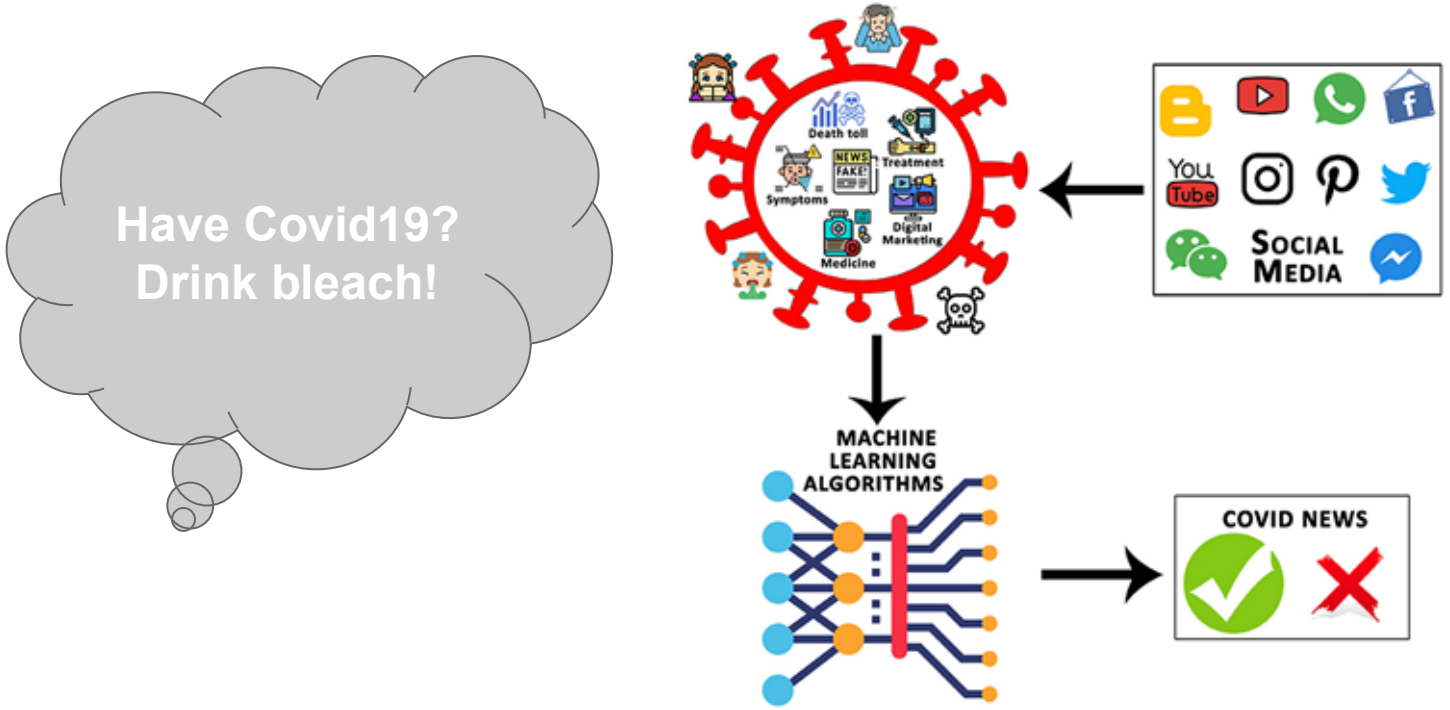
Authorship attribution: is the author male or female?

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

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 the 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

Fact verification: trustworthy or fake?



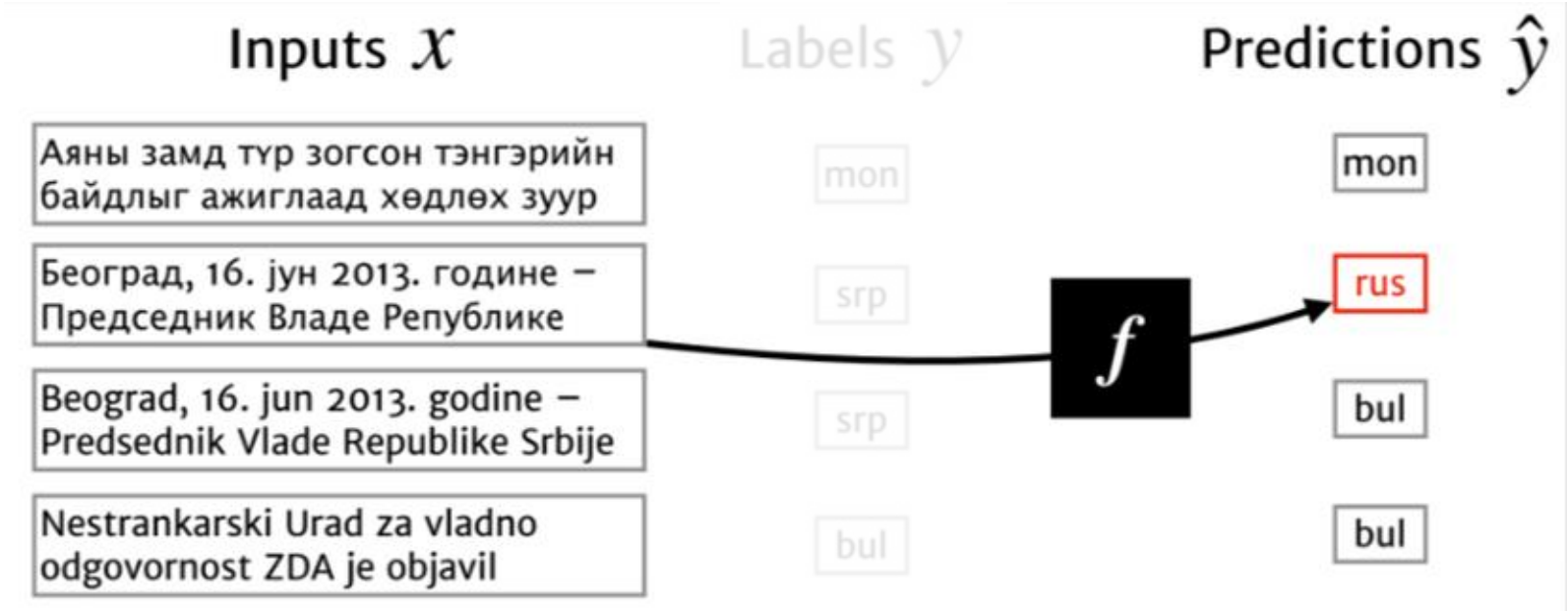
Detecting COVID-19-Related Fake News Using Feature Extraction

Suleman Khan, Saqib Hakak, N. Deepa, B. Prabadevi, Kapal Dev and Silvia Trelova

Text classification

- We might want to categorize the content of the text:
  - Spam detection (binary classification: spam/not spam)
  - Sentiment analysis (binary or multiway)
    - movie, restaurant, product reviews (pos/neg, or 1-5 stars)
    - political argument (pro/con, or pro/con/neutral)
    - Topic classification (multiway: sport/finance/travel/etc)
  - Language Identification (multiway: languages, language families)
  - …
- Or we might want to categorize the author of the text (authorship attribution)
  - Human- or machine generated?
  - Native language identification (e.g., to tailor language tutoring)
  - Diagnosis of disease (psychiatric or cognitive impairments)
  - Identification of gender, dialect, educational background, political orientation (e.g., in forensics [legal matters], advertising/marketing, campaigning, disinformation)
  - …

Text classification



Goal: create a function f that makes a prediction $\hat{y}$ given an input x

We will investigate:

1. How do we “digest” text into a form usable by a function?

   (Keywords for this section: features, feature extraction, feature selection, representations)

1. What kinds of strategies might we use to create our function *f*?

   (Keyword for this section: models)

1. How do we evaluate our function *f*?

   (Keyword for this section: … evaluation)

How do we “digest” text into a form usable by a function?

Classification: features (measurements)

- Perform measurements and obtain features

4.2, 212, 3.4, 1332

diameter, weight, softness, color

5.2, 315, 5.7, 4567

diameter, weight, softness, color

Text classification – feature extraction

What can we measure over text? Consider this movie review:

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 just to 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 before.

Text classification – feature extraction

What can we measure over text? Consider this movie review:

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 just to 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 before.

Bag-of-Words (BOW)

- Given a document d (e.g., a movie review) – how to represent d ?

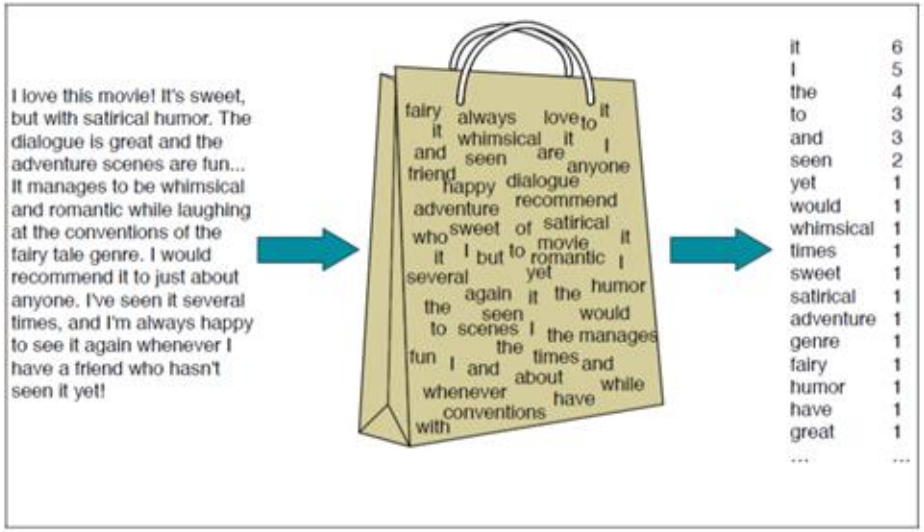


Figure 7.1 Intuition of the multinomial naive Bayes classifier applied to a movie review. The position of the words is ignored (the *bag of words* assumption) and we make use of the frequency of each word.

Figure from J&M 3rd ed. draft, sec 7.1

BOW feature extraction, independence assumption

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 just to 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 before.

(almost) the entire lexicon

word	count	relative frequency
love	10	0.0007
great	…	
recommend		
laugh		
happy		
…		
several		
boring		
…		

Types of textual features beyond BOW

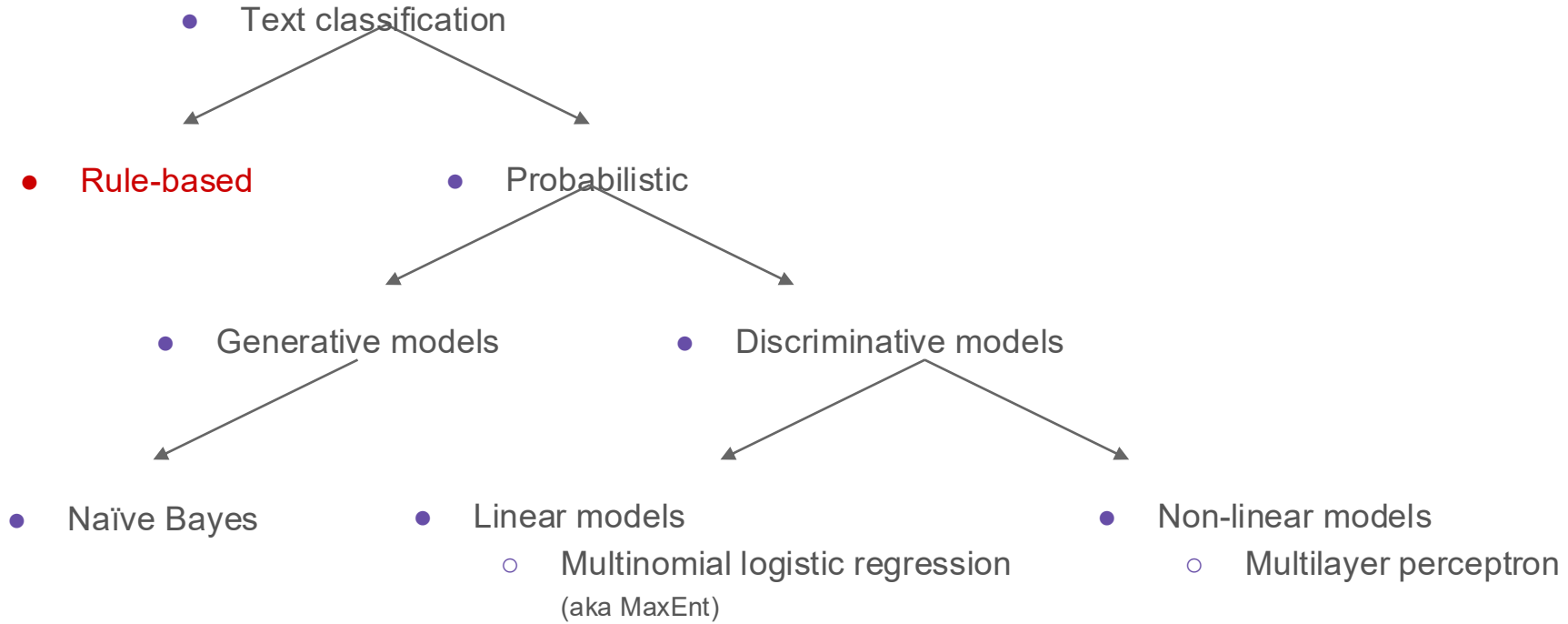
- Words
  - content words, stop-words
  - punctuation? tokenization? lemmatization? lowercase?
- Word sequences
  - bigrams, trigrams, n-grams
- Linguistic structure
  - Grammatical structure, sentence parse tree
  - Words' part-of-speech
- Word vectors
- …

Summary: Possible representations for text

- Bag-of-Words (BOW)
  - Easy, no effort required
  - Variable size, ignores sentential structure
- Hand-crafted features
  - Full control, can use NLP pipeline, class-specific features
  - Over-specific, incomplete, makes use of NLP pipeline
- Learned feature representations
  - Can learn to contain all relevant information
  - Needs to be learned

What kinds of strategies might we use to create our function *f*?

We'll consider alternative models for classification



Rule-based classifier

```
def classify_sentiment(document):
  for word in document:
    if word in {“good”, “wonderful”, “excellent”}:
      return 5
    if word in {“bad”, “awful”, “terrible”}:
      return 1
```

Rule-based classification: challenges

Sentiment: Half submarine flick, half ghost story, all in one a criminally neglected film.

Rule-based classification: challenges

Sentiment: Half submarine flick, half ghost story, all in one a criminally neglected film.

⟶ hard to identify a priori which words are informative (and what information they carry!)

Rule-based classification: challenges

Sentiment: Half submarine flick, half ghost story, all in one a criminally neglected film.

⟶ hard to identify a priori which words are informative (and what information they carry!)

Sentiment: It's not life-affirming, it's vulgar, it's mean, but I liked it.

Rule-based classification: challenges

Sentiment: Half submarine flick, half ghost story, all in one a criminally neglected film.

⟶ hard to identify a priori which words are informative (and what information they carry!)

Sentiment: It's not life-affirming, it's vulgar, it's mean, but I liked it.

⟶language pragmatics is complex to model at word level, word order (syntax) matters, but hard to encode in rules!

Rule-based classification: challenges

Sentiment: Half submarine flick, half ghost story, all in one a criminally neglected film.

⟶ hard to identify a priori which words are informative (and what information they carry!)

Sentiment: It's not life-affirming, it's vulgar, it's mean, but I liked it.

⟶language pragmatics is complex to model at word level, word order (syntax) matters, but hard to encode in rules!

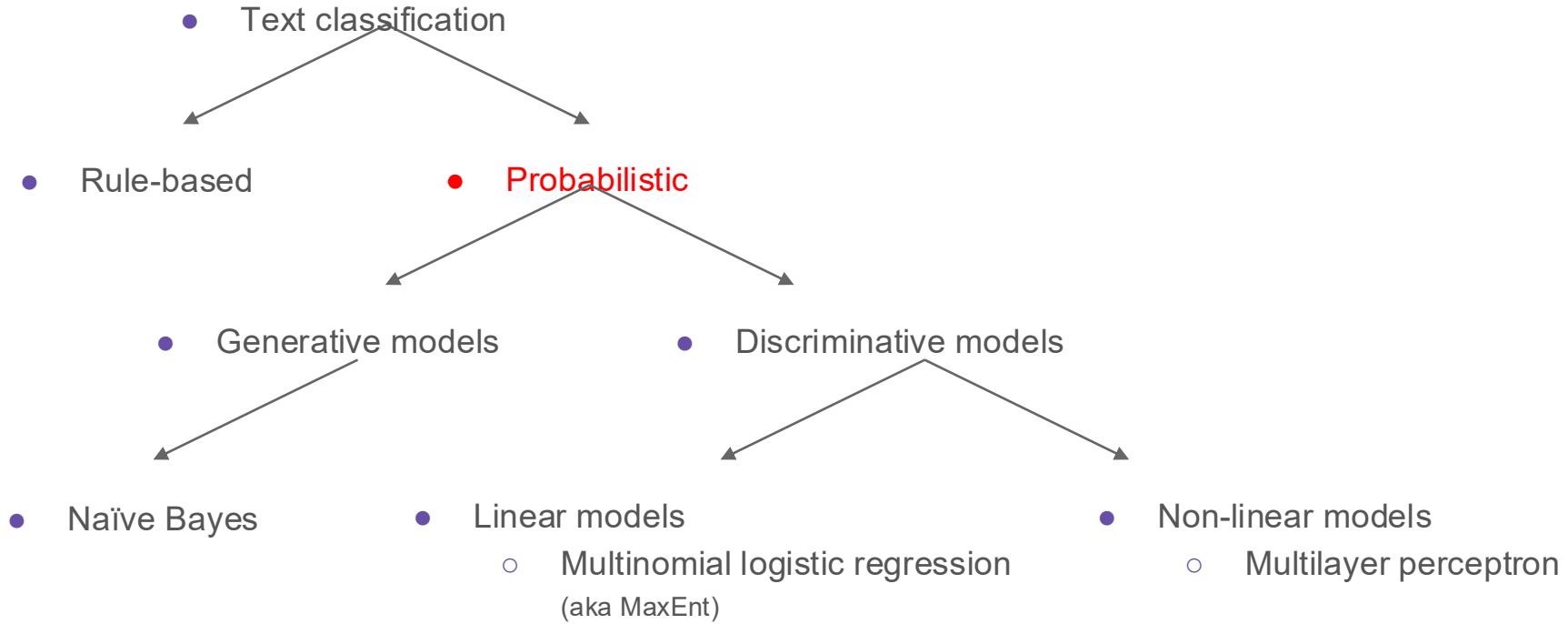
Language ID: All falter, stricken in kind.

⟶simple features can be misleading!

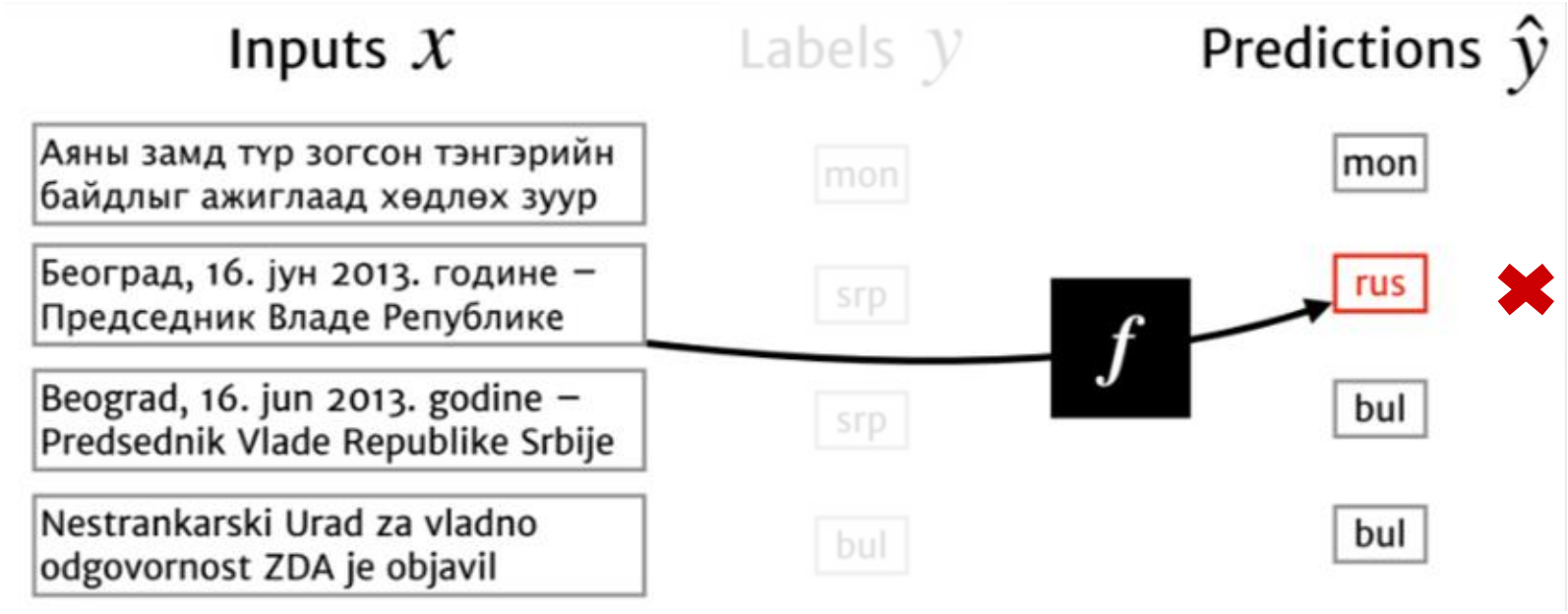
Rule-based classification

But don’t forget: if you don’t have access to data, speaker intuition and a bit of coding get you pretty far!

We'll consider alternative models for classification



Learning-based classification



pick the function f that does "best" on training data

Goal: ~~create a function f that makes a prediction $\hat{y}$ given an input x~~

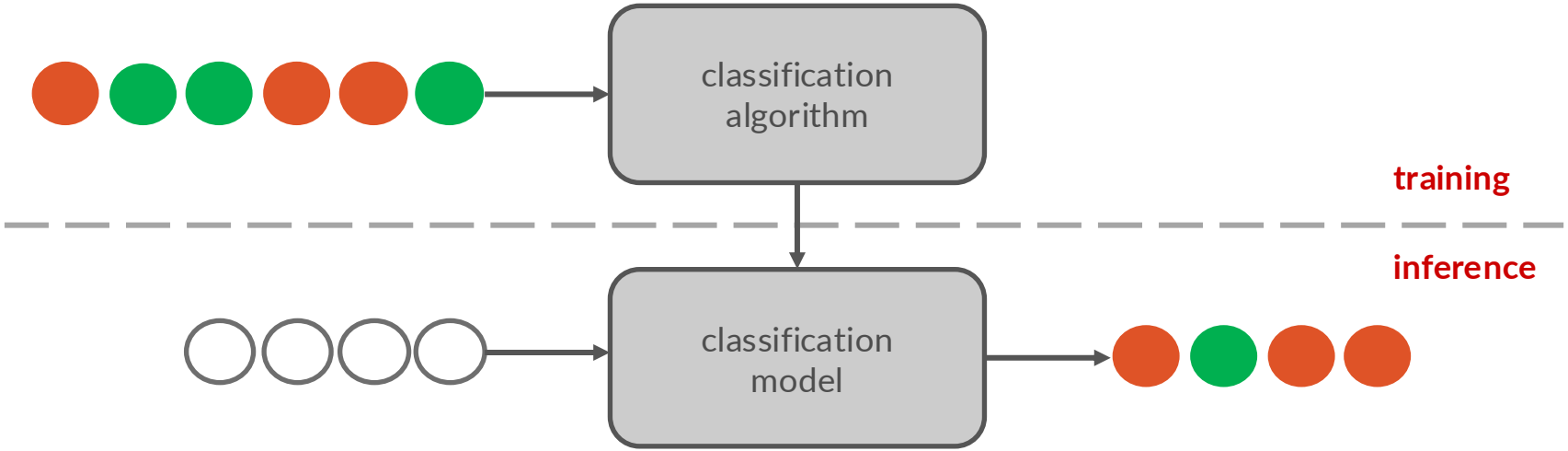
Classification: learning from data

- Supervised
  - labeled examples
    - Binary (true, false)
    - Multi-class classification (politics, sports, gossip)
    - Multi-label classification (#party #FRIDAY #fail)
- Unsupervised
  - no labeled examples
- Semi-supervised
  - labeled examples + non-labeled examples
- Weakly supervised
  - heuristically-labeled examples

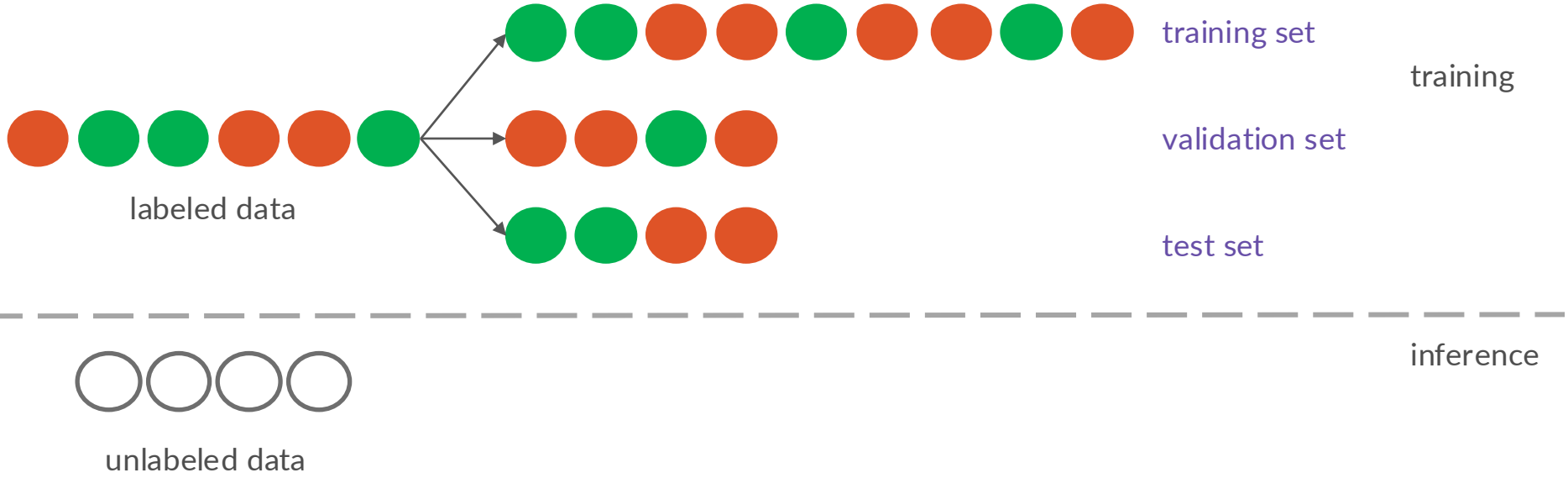
Where do datasets come from?

Human institutions	Noisy labels	Expert annotation	Crowd workers
Government proceedings	Domain names	Treebanks	Question answering
Product reviews	Link text	Biomedical corpora	Image captions

Supervised classification



Training, validation, and test sets



Supervised classification: formal setting

- Learn a classification model from labeled data on
  - properties ("features") and their importance ("weights")
- X: set of attributes or features $\{x_1, x_2, \ldots, x_n\}$
  - e.g. fruit measurements, or word counts extracted from an input documents
- y: a "class" label from the label set $Y = \{y_1, y_2, \ldots, y_k\}$
  - e.g., fruit type, or spam/not spam, positive/negative/neutral

Supervised classification: formal setting

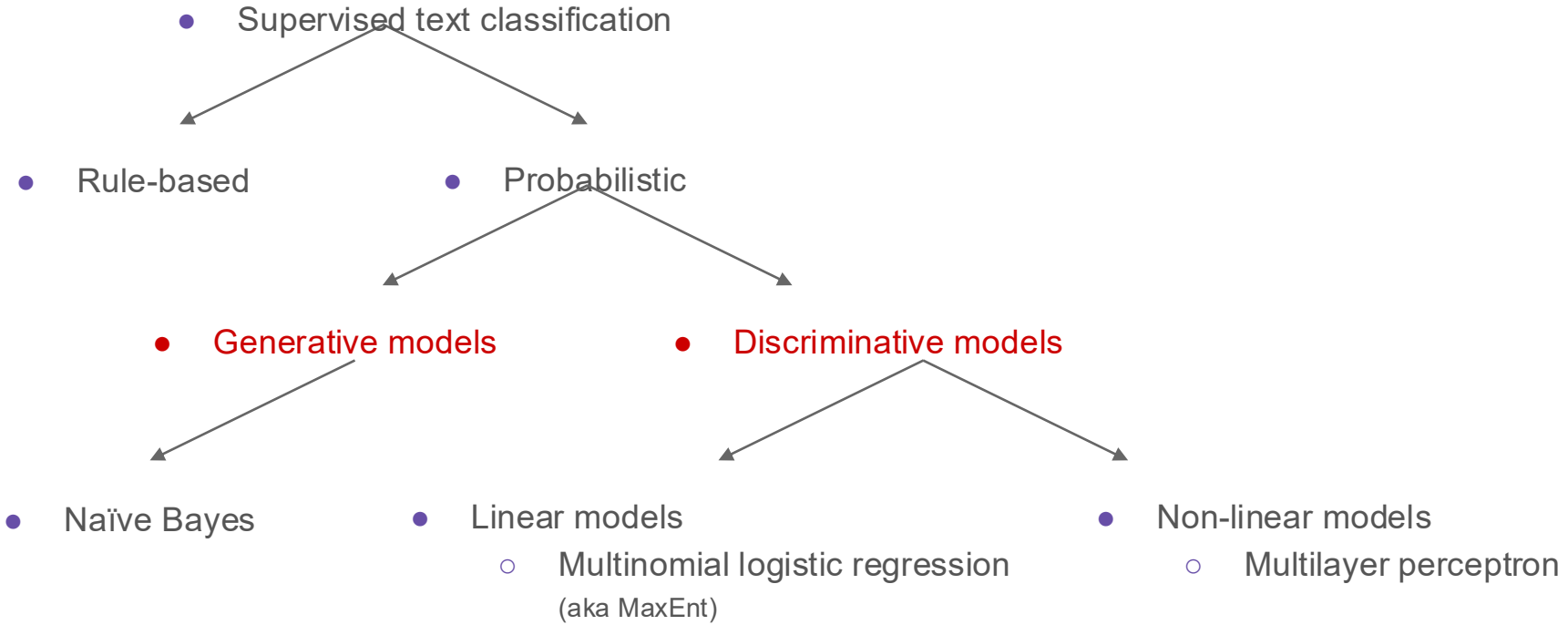
- Learn a classification model from labeled data on
  - properties ("features") and their importance ("weights")
- X: set of attributes or features $\{x_1, x_2, \ldots, x_n\}$
  - e.g. fruit measurements, or word counts extracted from an input documents
- y: a "class" label from the label set $Y = \{y_1, y_2, \ldots, y_k\}$
  - e.g., fruit type, or spam/not spam, positive/negative/neutral
- Given data samples $\{x_1, x_2, \ldots, x_n\}$ and corresponding labels $Y = \{y_1, y_2, \ldots, y_k\}$
- We train a function $f: x \in X \rightarrow y \in Y$ (the model)

Supervised classification: formal setting

- Learn a classification model from labeled data on
  - properties ("features") and their importance ("weights")
- X: set of attributes or features $\{x_1, x_2, \ldots, x_n\}$
  - e.g. fruit measurements, or word counts extracted from an input documents
- y: a "class" label from the label set $Y = \{y_1, y_2, \ldots, y_k\}$
  - e.g., fruit type, or spam/not spam, positive/negative/neutral

- At inference time, apply the model on new instances to predict the label $\hat{y}_i$

We'll consider alternative models for classification



Generative and discriminative models

- Generative model: a model that calculates the probability of the input data itself

$$P(X, Y)$$

joint

- Discriminative model: a model that calculates the probability of a latent trait given the data

$$P(Y \mid X)$$

conditional

Generative and discriminative models

imagenet

imagenet

Generative model

- Build a model of what's in a cat image
  - Knows about whiskers, ears, eyes
  - Assigns a probability to any image:
  - how cat-y is this image?
- Also build a model for dog images

Now given a new image:

Run both models and see which one fits better

Discriminative model

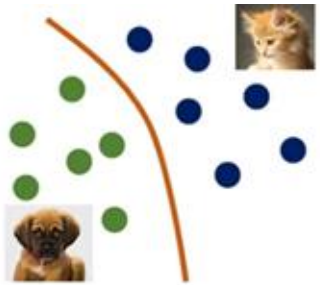
Just try to distinguish dogs from cats

Oh look, dogs have collars! Let's ignore everything else

Generative vs discriminative models

Learns the input distribution
Maximizes the joint probability: P(X, Y)
Estimates P(X|Y) to find P(Y|X) using Bayes' rule
Can generate new data
Typically, they are NOT used to solve classification tasks
Generative models possess discriminative properties

Hidden Markov Models | Naive Bayes | Gaussian Mixture Models | Gaussian Discriminant Analysis | LDA | Bayesian Networks



Learns the decision boundary between classes
Maximizes the conditional probability: P(Y|X)
Directly estimates P(Y|X)
Cannot generate new data
Specifically meant for classification tasks
Discriminative models don't possess generative properties

Logistic Regression | Random Forests | SVMs | Neural Networks | Decision Tree | kNN

https://blog.dailydoseofds.com/p/an-intuitive-guide-to-generative
https://medium.com/@jordi299/about-generative-and-discriminative-models-d8958b67ad32

Generative and discriminative models

- Generative text classification: Learn a model of the joint P(X, y), and find

$$\hat{y} = \underset{\tilde{y}}{\text{argmax}}\ P(X, \tilde{y})$$

- Discriminative text classification: Learn a model of the conditional P(y | X), and find

$$\hat{y} = \underset{\tilde{y}}{\text{argmax}}\ P(\tilde{y}|X)$$

Finding the correct class c from a document d in Generative vs Discriminative Classifiers

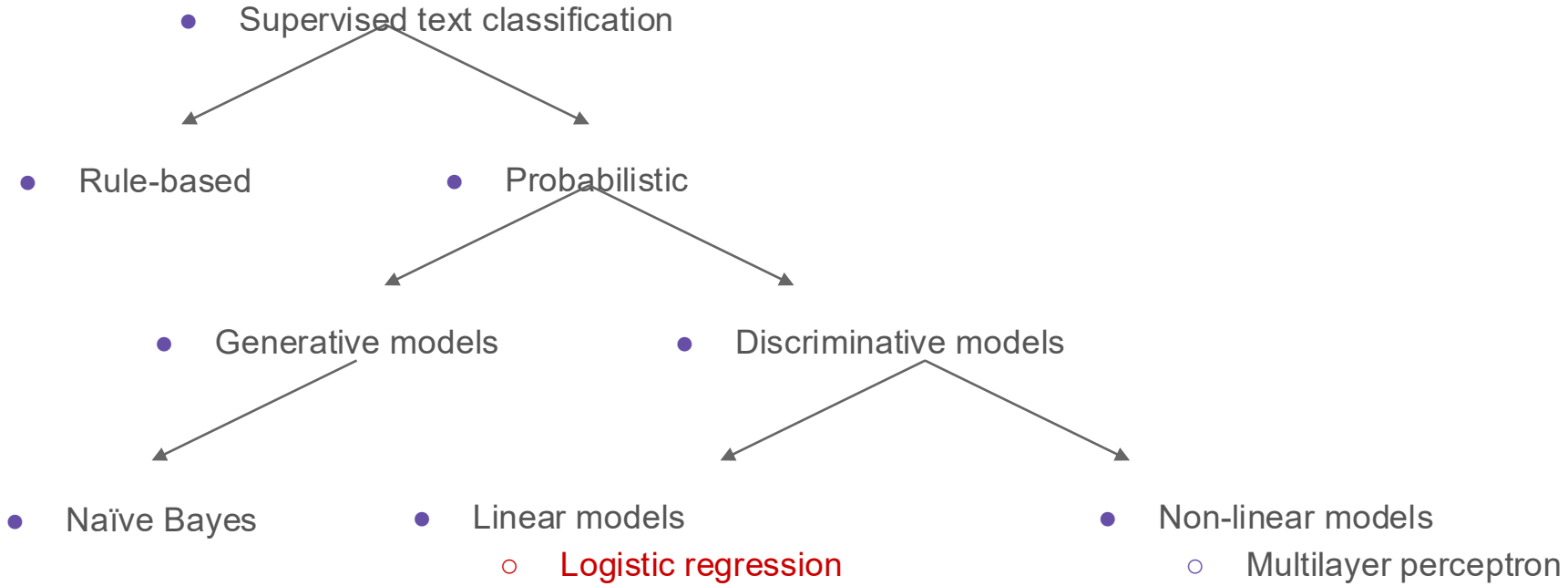
- Naive Bayes

$$\hat{c} = \underset{c \in C}{\text{argmax}} \overbrace{P(d|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}}$$

- Logistic Regression

$$\hat{c} = \underset{c \in C}{\text{argmax}} \overbrace{P(c|d)}^{\text{posterior}}$$

Logistic regression



Logistic regression classifier

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

Text classification

Input:

- a document d (e.g., a movie review)
- a fixed set of classes $C = \{c_1, c_2, \ldots c_j\}$ (e.g., positive, negative, neutral)

Output

- a predicted class $\hat{y} \in C$

Binary 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 $\{x_1, x_2, \ldots, x_n\}$
  - We compute an output: a predicted class $\hat{y}^{(i)} \in \{0,1\}$

Features in logistic regression

- For feature $x_i \in \{x_1, x_2, \ldots, x_n\}$, weight $w_i \in \{w_1, w_2, \ldots, w_n\}$ tells us how important is x_i
  - x_i = "review contains 'awesome'": w_i = +10
  - x_j = "review contains horrible": w_j = -10
  - x_k = "review contains 'mediocre'": w_k = -2

Logistic Regression for one observation x

- Input observation: vector $x^{(i)} = \{x_1, x_2, \ldots, x_n\}$
- Weights: one per feature: $W = [w_1, w_2, \ldots, w_n]$
  - Sometimes we call the weights $\theta = [\theta_1, \theta_2, \ldots, \theta_n]$
- Output: a predicted class $\hat{y}^{(i)} \in \{0,1\}$

multinomial logistic regression: $\hat{y}^{(i)} \in \{0,1, 2, 3, 4\}$

How to do classification

- For each feature x_i, weight w_i tells us importance of x_i
  - (Plus we'll have a bias b)
  - We'll sum up all the weighted features and the bias

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

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

If this sum is high, we say $y=1$; if low, then $y=0$

But we want a probabilistic classifier

We need to formalize “sum is high”

- We’d like a principled classifier that gives us a probability… why?
- We want a model that can tell us:
  - $p(y=1|x; \theta)$
  - $p(y=0|x; \theta)$

The problem: z isn't a probability, it's just a number!

- z ranges from $-\infty$ to ∞

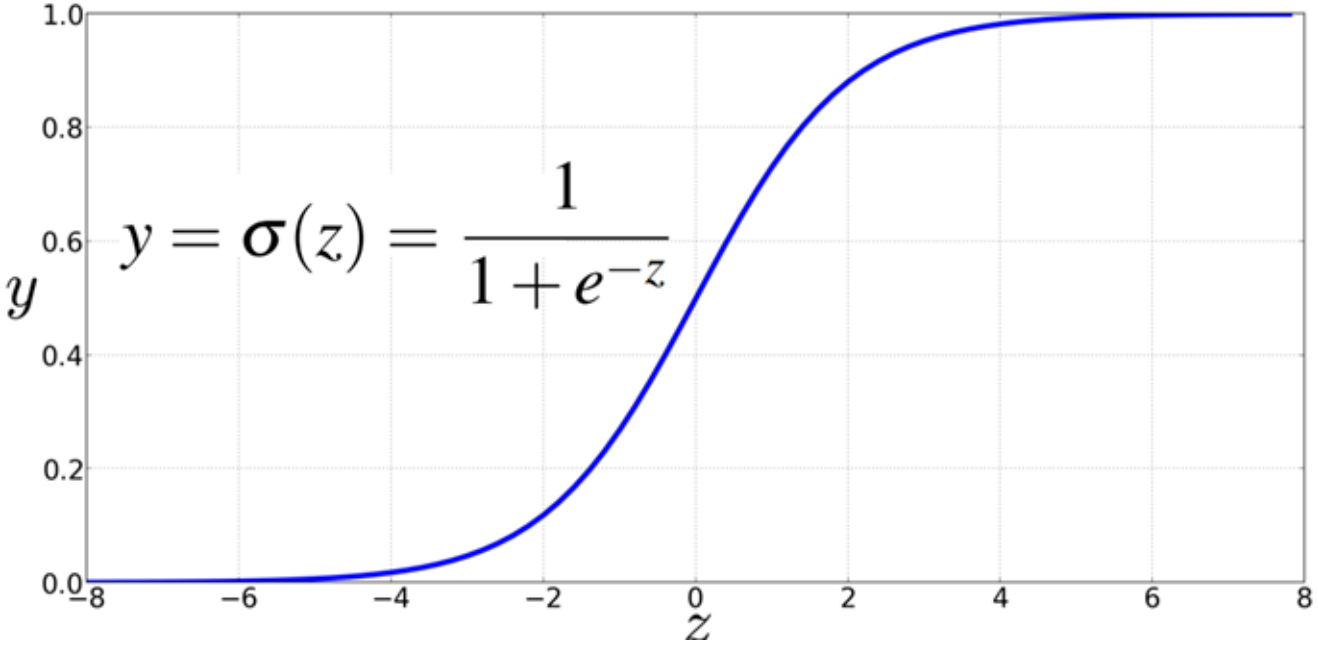
$$z = w \cdot x + b$$

- Solution: use a function of z that goes from 0 to 1

"sigmoid" or
"logistic" function

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

The very useful sigmoid or logistic function



Idea of logistic regression

- We'll compute w·x+b
- And then we'll pass it through the sigmoid function:

$$\sigma(w \cdot x + b)$$

- And we'll just treat it as a probability

Making probabilities with sigmoids

$$
\begin{aligned}
P(y=1) &= \sigma(w \cdot x + b) \\
&= \frac{1}{1+\exp\left(-(w \cdot x + b)\right)}
\end{aligned}
$$

Making probabilities with sigmoids

$$
\begin{aligned}
P(y=1) &= \sigma(w \cdot x + b) \\
&= \frac{1}{1+\exp\left(-(w \cdot x + b)\right)}
\end{aligned}
$$

$$
\begin{aligned}
P(y=0) &= 1 - \sigma(w \cdot x + b) \\
&= 1 - \frac{1}{1+\exp\left(-(w \cdot x + b)\right)} \\
&= \frac{\exp\left(-(w \cdot x + b)\right)}{1+\exp\left(-(w \cdot x + b)\right)}
\end{aligned}
$$

By the way:

$$
\begin{aligned}
P(y=0) &= 1-\sigma(w\cdot x+b) &&= \sigma(-(w\cdot x+b)) \\
&= 1-\frac{1}{1+\exp(-(w\cdot x+b))} && \text{Because} \\
&= \frac{\exp(-(w\cdot x+b))}{1+\exp(-(w\cdot x+b))} && 1-\sigma(x)=\sigma(-x)
\end{aligned}
$$

Turning a probability into a classifier

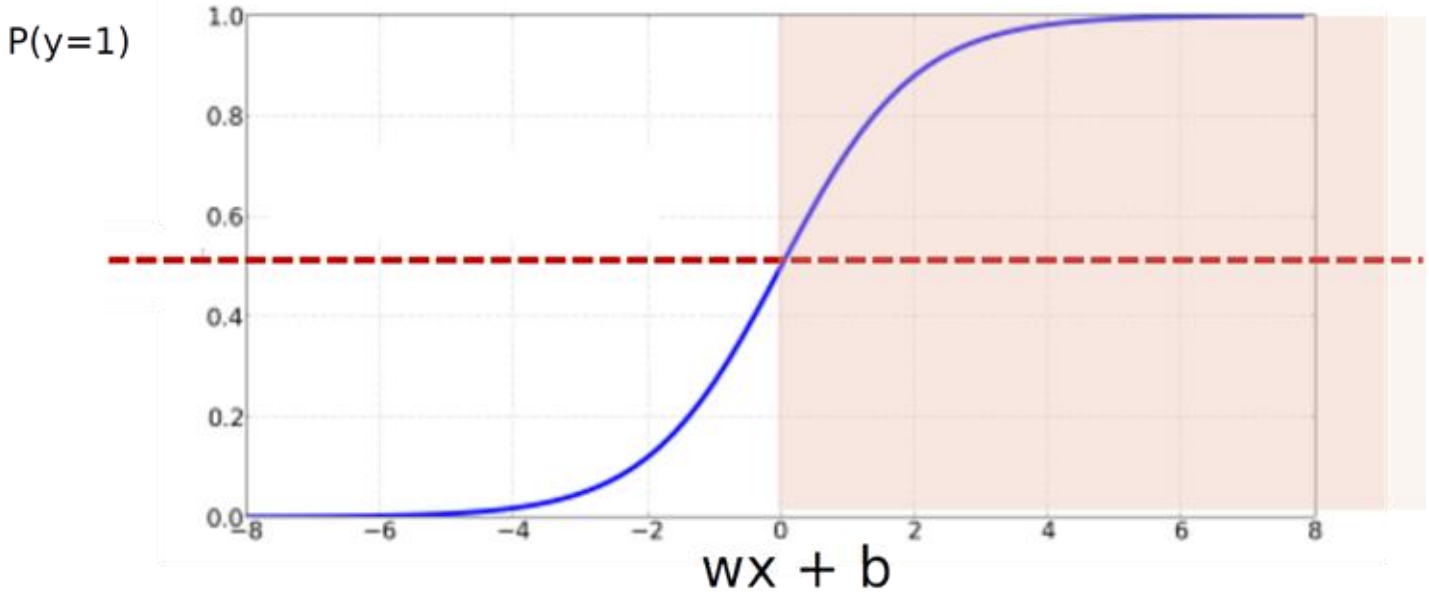
$$\hat{y} = \begin{cases} 1 & \text{if } P(y=1|x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

- 0.5 here is called the **decision boundary**

The probabilistic classifier

$$P(y=1) = \sigma(w \cdot x + b)$$

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



Turning a probability into a classifier

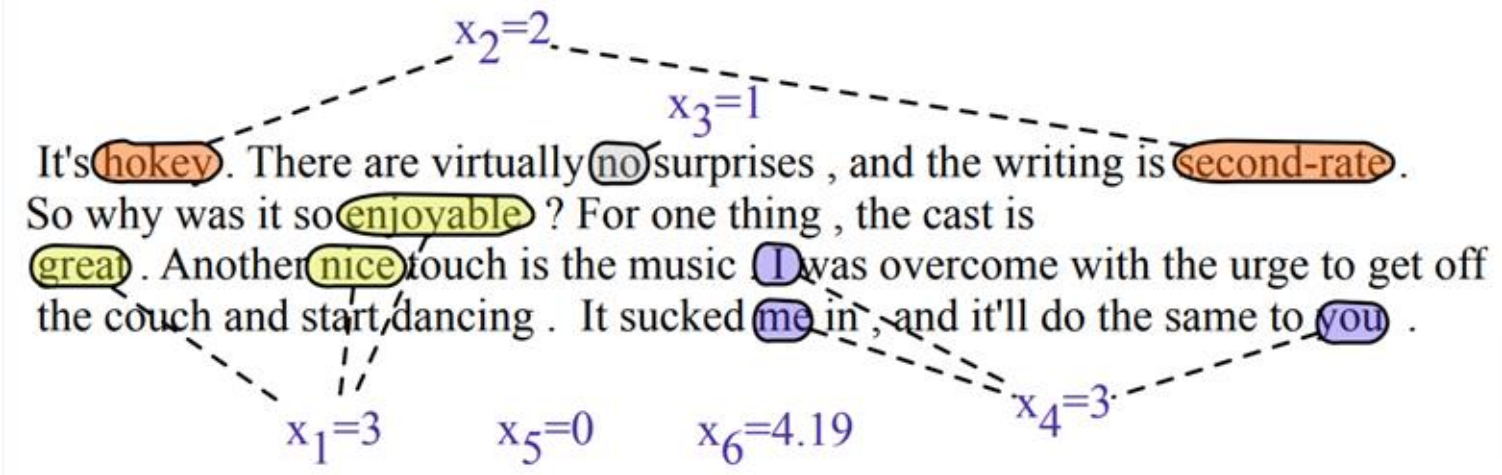
$$\hat{y} = \begin{cases} 1 & \text{if } P(y=1|x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

if $w \cdot x + b > 0$

if $w \cdot x + b \leq 0$

Sentiment example: does y=1 or y=0?

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 .



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)	$\ln(66) = 4.19$

Classifying sentiment for input x

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)	$\ln(66) = 4.19$

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

$b = 0.1$

Classifying sentiment for input x

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

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

Scaling input features

- z-score

$$\mu_i = \frac{1}{m}\sum_{j=1}^{m} x_i^{(j)} \qquad \sigma_i = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(\mathbf{x}_i^{(j)} - \mu_i\right)}$$

$$\mathbf{x}'_i = \frac{\mathbf{x}_i - \mu_i}{\sigma_i}$$

- normalize

$$\mathbf{x}'_i = \frac{\mathbf{x}_i - \min(\mathbf{x}_i)}{\max(\mathbf{x}_i) - \min(\mathbf{x}_i)}$$

Wait, where did the W's come from?

- Supervised classification:
  - A training time we know the correct label y (either 0 or 1) for each x.
  - But what the system produces at inference time is an estimate $\hat{y}$

Wait, where did the W's come from?

- Supervised classification:
  - A training time we know the correct label y (either 0 or 1) for each x.
  - But what the system produces at inference time is an estimate $\hat{y}$

- We want to set w and b to minimize the **distance** between our estimate $\hat{y}^{(i)}$ and the true $y^{(i)}$
  - 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

Components of a probabilistic machine learning classifier

Given m input/output pairs $(x^{(i)}, y^{(i)})$:

1. A **feature representation** for the input. For each input observation $x^{(i)}$, a vector of features $[x_1, x_2, \ldots, x_n]$. Feature j for input $x^{(i)}$ is x_j, more completely $x_1^{(i)}$, or sometimes $f_j(x)$.

2. A **classification function** that computes $\hat{y}$ the estimated class, via $p(y|x)$, like the **sigmoid** functions

3. An **objective function** for learning, like **cross-entropy loss**

4. An algorithm for **optimizing** the objective function: **stochastic gradient descent**

Learning components in LR

A loss function:

- **cross-entropy loss**

An optimization algorithm:

- **stochastic gradient descent**

Loss function: the distance between ŷ and y

We want to know how far is the classifier output $\hat{y} = \sigma(w \cdot x + b)$

from the true output: y [= either 0 or 1]

We'll call this difference: $L(\hat{y},y)$ = how much $\hat{y}$ differs from the true y

An objective function or loss

We want loss to be:

- smaller if the model estimate $\hat{y}$ is close to correct
- bigger if model is confused

Deriving cross-entropy loss for a single observation x

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

Since there are only 2 discrete outcomes (0 or 1) we can express the probability $p(y|x)$ from our classifier (the thing we want to maximize) as

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

Deriving cross-entropy loss for a single observation x

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

Since there are only 2 discrete outcomes (0 or 1) we can express the probability $p(y|x)$ from our classifier (the thing we want to maximize) as

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

Noting:

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

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

Deriving cross-entropy loss for a single observation x

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

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

Deriving cross-entropy loss for a single observation x

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

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

Now take the log of both sides (mathematically handy)

Maximize: $\log p(y|x) = \log \left[\hat{y}^y (1-\hat{y})^{1-y}\right]$

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

Deriving cross-entropy loss for a single observation x

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

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

Now take the log of both sides (mathematically handy)

Maximize: $\log p(y|x) = \log \left[\hat{y}^y (1-\hat{y})^{1-y}\right]$

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

Whatever values maximize $\log p(y|x)$ will also maximize $p(y|x)$

Deriving cross-entropy loss for a single observation x

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

Maximize: $$\begin{aligned}\log p(y|x) &= \log\left[\hat{y}^y(1-\hat{y})^{1-y}\right] \\ &= y\log\hat{y} + (1-y)\log(1-\hat{y})\end{aligned}$$

Now flip sign to turn this into a loss: something to minimize

Deriving cross-entropy loss for a single observation x

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

Maximize: $$\begin{aligned}\log p(y|x) &= \log\left[\hat{y}^y(1-\hat{y})^{1-y}\right] \\ &= y\log\hat{y} + (1-y)\log(1-\hat{y})\end{aligned}$$

Now flip sign to turn this into a loss: something to minimize

Minimize: $$L_{\text{CE}}(\hat{y},y) = -\log p(y|x) = -[y\log\hat{y} + (1-y)\log(1-\hat{y})]$$

Deriving cross-entropy loss for a single observation x

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

Maximize: $$\begin{aligned}\log p(y|x) &= \log\left[\hat{y}^y(1-\hat{y})^{1-y}\right] \\ &= y\log\hat{y} + (1-y)\log(1-\hat{y})\end{aligned}$$

Now flip sign to turn this into a **cross-entropy loss**: something to minimize

Minimize: $$L_{\text{CE}}(\hat{y},y) = -\log p(y|x) = -[y\log\hat{y} + (1-y)\log(1-\hat{y})]$$

Deriving cross-entropy loss for a single observation x

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

Maximize: $$\begin{aligned}\log p(y|x) &= \log\left[\hat{y}^y(1-\hat{y})^{1-y}\right] \\ &= y\log\hat{y} + (1-y)\log(1-\hat{y})\end{aligned}$$

Now flip sign to turn this into a **cross-entropy loss**: something to minimize

Minimize: $$L_{\text{CE}}(\hat{y},y) = -\log p(y|x) = -[y\log\hat{y} + (1-y)\log(1-\hat{y})]$$

Or, plug in definition of $\hat{y} = \sigma(w\cdot x+b)$

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

Let's see if this works for our sentiment example

We want loss to be:

- smaller if the model estimate $\hat{y}$ is close to correct
- bigger if model is confused

Let's first suppose the true label of this is $y=1$ (positive)

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 .

Let's see if this works for our sentiment example

True value is y=1 (positive). How well is our model doing?

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

Pretty well!

Let's see if this works for our sentiment example

True value is y=1 (positive). How well is our model doing?

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

Pretty well! What's the loss?

$$\begin{aligned} L_{\text{CE}}(\hat{y}, y) &= -[y \log \sigma(\mathbf{w} \cdot \mathbf{x} + b) + (1-y) \log (1 - \sigma(\mathbf{w} \cdot \mathbf{x} + b))] \\ &= -[\log \sigma(\mathbf{w} \cdot \mathbf{x} + b)] \\ &= -\log(.70) \\ &= .36 \end{aligned}$$

Let's see if this works for our sentiment example

Suppose the true value instead was y=0 (negative).

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

Let's see if this works for our sentiment example

Suppose the true value instead was y=0 (negative).

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

What's the loss?

$$\begin{aligned} L_{\text{CE}}(\hat{y},y) = & -[y\log\sigma(\mathbf{w}\cdot\mathbf{x}+b)+(1-y)\log(1-\sigma(\mathbf{w}\cdot\mathbf{x}+b))] \\ = & -[\log(1-\sigma(\mathbf{w}\cdot\mathbf{x}+b))] \\ = & -\log(.30) \\ = & 1.2 \end{aligned}$$

Let's see if this works for our sentiment example

The loss when the model was right (if true y=1)

$$\begin{aligned} L_{\text{CE}}(\hat{y}, y) &= -[y \log \sigma(\mathbf{w} \cdot \mathbf{x} + b) + (1-y) \log (1 - \sigma(\mathbf{w} \cdot \mathbf{x} + b))] \\ &= -[\log \sigma(\mathbf{w} \cdot \mathbf{x} + b)] \\ &= -\log(.70) \\ &= .36 \end{aligned}$$

The loss when the model was wrong (if true y=0)

$$\begin{aligned} L_{\text{CE}}(\hat{y}, y) &= -[y \log \sigma(\mathbf{w} \cdot \mathbf{x} + b) + (1-y) \log (1 - \sigma(\mathbf{w} \cdot \mathbf{x} + b))] \\ &= -[\log (1 - \sigma(\mathbf{w} \cdot \mathbf{x} + b))] \\ &= -\log (.30) \\ &= 1.2 \end{aligned}$$

Sure enough, loss was bigger when model was wrong!

Learning components

A loss function:

- **cross-entropy loss**

An optimization algorithm:

- **stochastic gradient descent**

Stochastic Gradient Descent

- Stochastic Gradient Descent algorithm
  - is used to optimize the weights
  - for logistic regression
  - also for neural networks

Our goal: minimize the loss

Let's make explicit that the loss function is parameterized by weights $\theta=(w,b)$

- And we'll represent $\hat{y}$ as $f(x; \theta)$ to make the dependence on θ more obvious

We want the weights that minimize the loss, averaged over all examples:

$$\hat{\theta} = \underset{\theta}{\text{argmin}} \frac{1}{m} \sum_{i=1}^{m} L_{\text{CE}}(f(x^{(i)}; \theta), y^{(i)})$$

$$L_{\text{CE}}(\hat{y}, y)$$

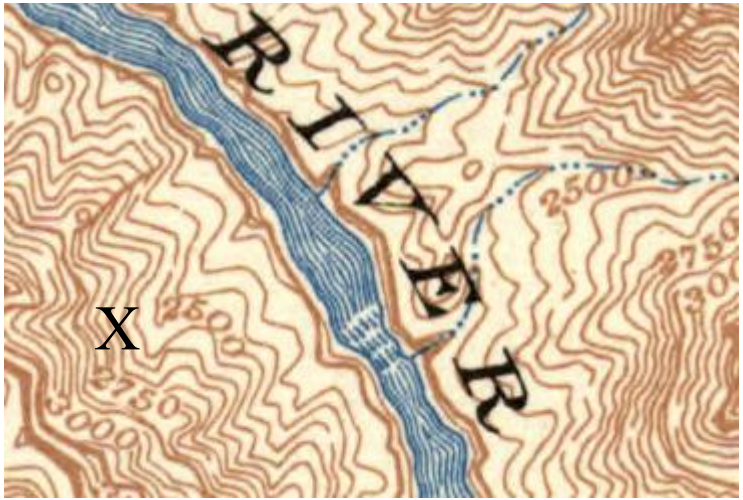
Intuition of gradient descent

How do I get to the bottom of this river canyon?

Look around me 360∘

Find the direction of steepest slope down

Go that way



Our goal: minimize the loss

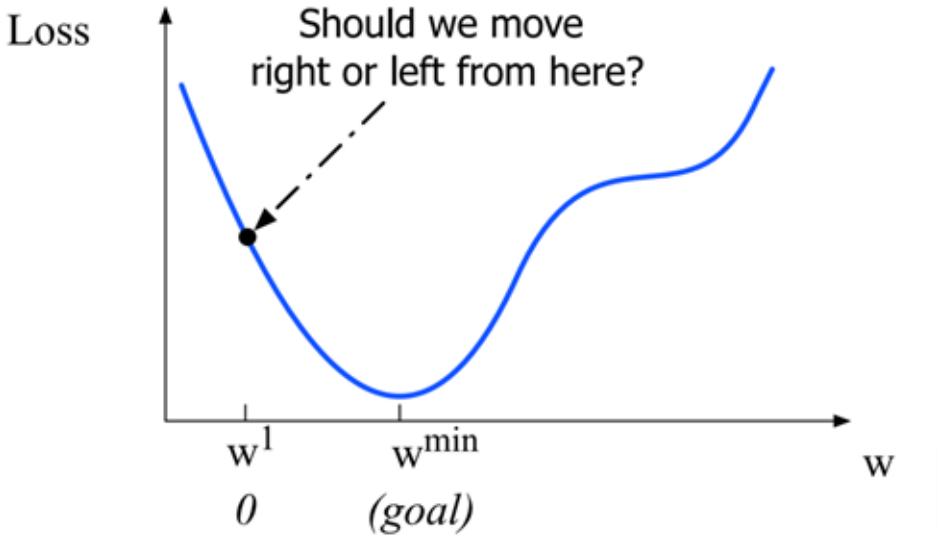
For logistic regression, loss function is **convex**

- A convex function has just one minimum
- Gradient descent starting from any point is guaranteed to find the minimum
  - (Loss for neural networks is non-convex)

Let's first visualize for a single scalar w

Q: Given current w, should we make it bigger or smaller?

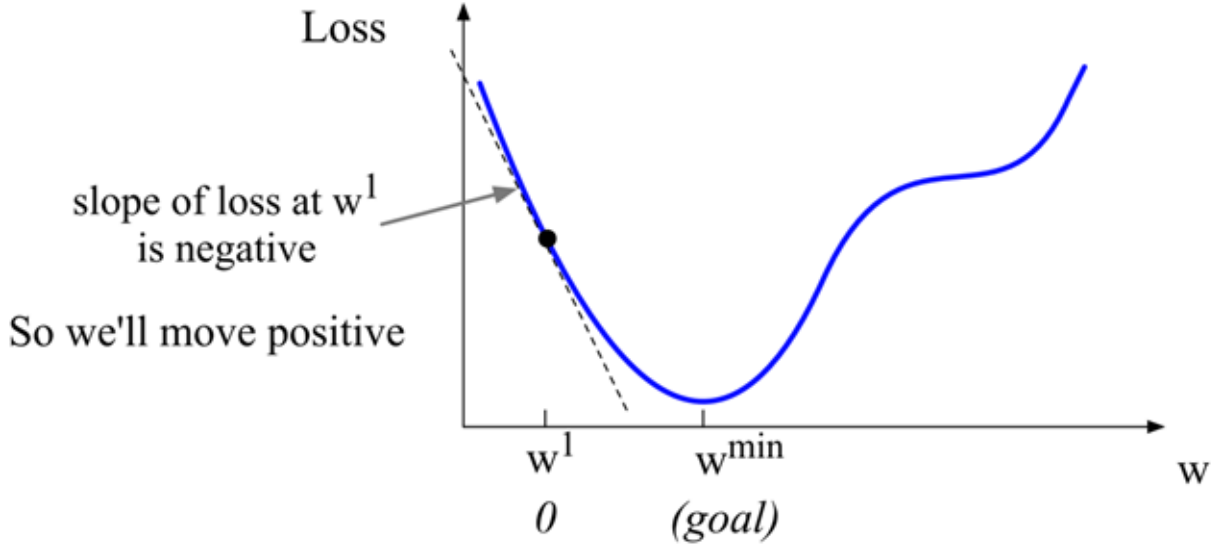
A: Move w in the reverse direction from the slope of the function



Let's first visualize for a single scalar w

Q: Given current w, should we make it bigger or smaller?

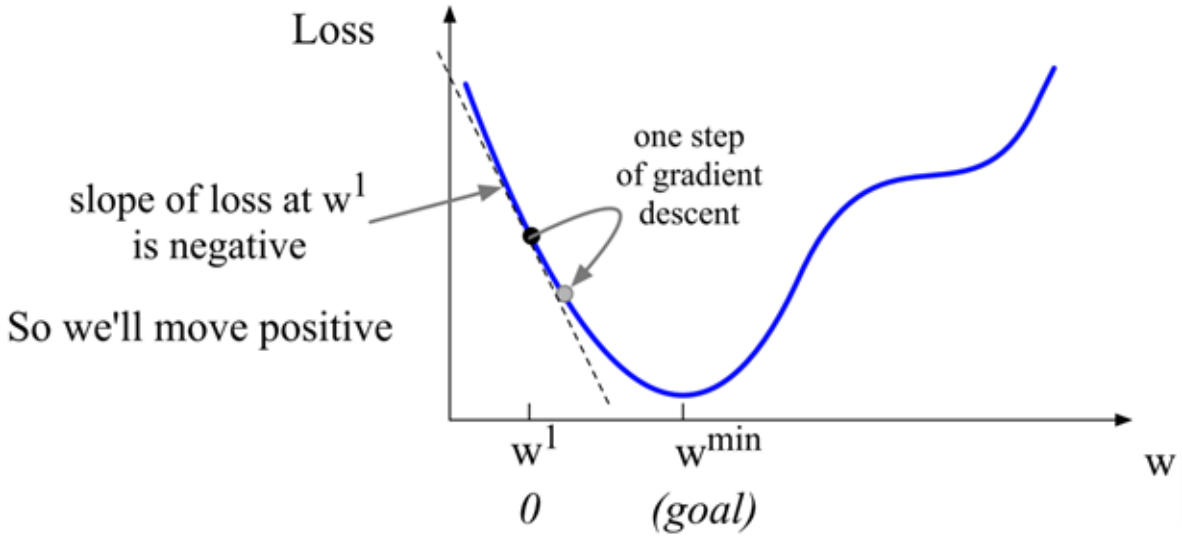
A: Move w in the reverse direction from the slope of the function



Let's first visualize for a single scalar w

Q: Given current w, should we make it bigger or smaller?

A: Move w in the reverse direction from the slope of the function



Gradients

The **gradient** of a function of many variables is a vector pointing in the direction of the greatest increase in a function.

Gradient Descent: Find the gradient of the loss function at the current point and move in the **opposite** direction.

How much do we move in that direction?

- The value of the gradient (slope in our example) $\frac{d}{dw}L(f(x;w),y)$
  - weighted by a learning rate η

- Higher learning rate means move w faster

$$w^{t+1} = w^t - \eta \frac{d}{dw} L(f(x;w),y)$$

Now let's consider N dimensions

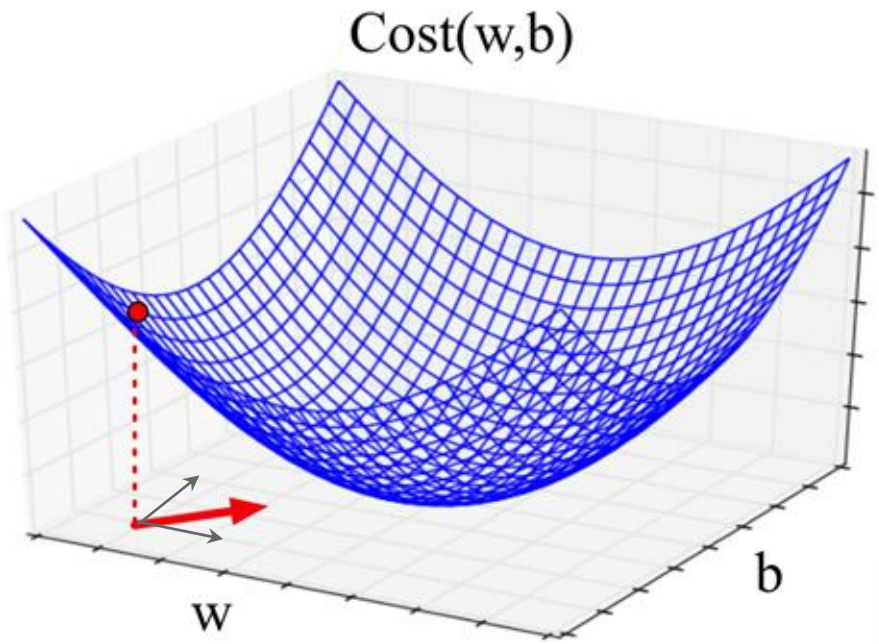
We want to know where in the N-dimensional space (of the N parameters that make up θ) we should move.

The gradient is just such a vector; it expresses the directional components of the sharpest slope along each of the N dimensions.

Imagine 2 dimensions, w and b

Visualizing the gradient vector at the red point

It has two dimensions shown in the x-y plane



Real gradients

Are much longer; lots and lots of weights

For each dimension w_i the gradient component i tells us the slope with respect to that variable.

- "How much would a small change in w_i influence the total loss function L?"
- We express the slope as a partial derivative ∂ of the loss ∂w_i $\frac{\partial}{\partial w_i}$

The gradient is then defined as a vector of these partials.

The gradient

We'll represent $\hat{y}$ as $f(x; \theta)$ to make the dependence on θ more obvious:

$$\nabla_\theta L(f(x;\theta),y)) = \begin{bmatrix} \frac{\partial}{\partial w_1} L(f(x;\theta),y) \\ \frac{\partial}{\partial w_2} L(f(x;\theta),y) \\ \vdots \\ \frac{\partial}{\partial w_n} L(f(x;\theta),y) \end{bmatrix}$$

The final equation for updating θ based on the gradient is thus:

$$\theta_{t+1} = \theta_t - \eta \nabla L(f(x;\theta),y)$$

What are these partial derivatives for logistic regression?

The loss function

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

The elegant derivative of this function (see Section 5.10 for the derivation)

$$\begin{aligned} \frac{\partial L_{\text{CE}}(\hat{y}, y)}{\partial w_j} &= [\sigma(w \cdot x + b) - y] x_j \\ &= (\hat{y} - y) \mathbf{x}_j \end{aligned}$$

function STOCHASTIC GRADIENT DESCENT($L()$, $f()$, x, y) **returns** θ

where: L is the loss function
f is a function parameterized by θ
x is the set of training inputs $x^{(1)}$, $x^{(2)}$,..., $x^{(m)}$
y is the set of training outputs (labels) $y^{(1)}$, $y^{(2)}$,..., $y^{(m)}$

$\theta \leftarrow 0$

repeat til done

For each training tuple $(x^{(i)}, y^{(i)})$ (in random order)

1. Optional (for reporting): # How are we doing on this tuple?
   Compute $\hat{y}^{(i)} = f(x^{(i)}; \theta)$ # What is our estimated output $\hat{y}$?
   Compute the loss $L(\hat{y}^{(i)}, y^{(i)})$ # How far off is $\hat{y}^{(i)}$) from the true output $y^{(i)}$?
2. $g \leftarrow \nabla_\theta L(f(x^{(i)}; \theta), y^{(i)})$ # How should we move θ to maximize loss?
3. $\theta \leftarrow \theta - \eta\, g$ # Go the other way instead

return θ

Hyperparameters

The learning rate η is a **hyperparameter**

- too high: the learner will take big steps and overshoot
- too low: the learner will take too long

Hyperparameters:

- Briefly, a special kind of parameter for an ML model
- Instead of being learned by algorithm from supervision (like regular parameters), they are chosen by algorithm designer.

Mini-batch training

Stochastic gradient descent chooses a single random example at a time.

That can result in choppy movements

More common to compute gradient over batches of training instances.

Batch training: entire dataset

Mini-batch training: m examples (512, or 1024)

Overfitting

A model that perfectly match the training data has a problem.

It will also **overfit** to the data, modeling noise

- A random word that perfectly predicts y (it happens to only occur in one class) will get a very high weight.
- Failing to generalize to a test set without this word.

A good model should be able to **generalize**

Regularization

A solution for overfitting

Add a **regularization** term $R(\theta)$ to the loss function (for now written as maximizing logprob rather than minimizing loss)

$$\hat{\theta} = \underset{\theta}{\text{argmax}} \sum_{i=1}^{m} \log P(y^{(i)}|x^{(i)}) - \alpha R(\theta)$$

Idea: choose an $R(\theta)$ that penalizes large weights

- fitting the data well with lots of big weights not as good as fitting the data a little less well, with small weights

L2 regularization (ridge regression)

The sum of the squares of the weights

$$R(\theta) = ||\theta||_2^2 = \sum_{j=1}^{n} \theta_j^2$$

L2 regularized objective function:

$$\hat{\theta} = \underset{\theta}{\text{argmax}} \left[\sum_{i=1}^{m} \log P(y^{(i)}|x^{(i)}) \right] - \alpha \sum_{j=1}^{n} \theta_j^2$$

L1 regularization (=lasso regression)

The sum of the (absolute value of the) weights

$$R(\theta) = ||\theta||_1 = \sum_{i=1}^{n} |\theta_i|$$

L1 regularized objective function:

$$\hat{\theta} = \underset{\theta}{\text{argmax}} \left[\sum_{1=i}^{m} \log P(y^{(i)}|x^{(i)}) \right] - \alpha \sum_{j=1}^{n} |\theta_j|$$

Multinomial Logistic Regression

Often we need more than 2 classes

- Positive/negative/neutral
- Parts of speech (noun, verb, adjective, adverb, preposition, etc.)
- Classify emergency SMSs into different actionable classes

If >2 classes we use **multinomial logistic regression**

= Softmax regression
= Multinomial logit
= (defunct names : Maximum entropy modeling or MaxEnt)

So "logistic regression" will just mean binary (2 output classes)

Multinomial Logistic Regression

The probability of everything must still sum to 1

P(positive|doc) + P(negative|doc) + P(neutral|doc) = 1

Need a generalization of the sigmoid called the **softmax**

- Takes a vector $z = [z_1, z_2, ..., z_k]$ of k arbitrary values
- Outputs a probability distribution
- each value in the range $[0,1]$
- all the values summing to 1

We'll discuss it more when we talk about neural networks

One-hot representation

Gold labels – one-hot representations

$$[0, 0, ..., 1, 0, 0]$$

Predicted values – vector of class probabilities

$$[0.1, 0.05, ..., .08, 0, 0.07]$$

Sigmoid → softmax

$$\frac{1}{1+e^{-z}} \longrightarrow \frac{e^{z_j}}{\sum_{c=1}^{k} e^{z_c}}$$

The **softmax** function

- Turns a vector $z = [z_1, z_2, ..., z_k]$ of k arbitrary values (logits) into probabilities

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^{k} \exp(z_j)} \quad 1 \leq i \leq k$$

- The denominator $\sum_{i=1}^{k} e^{z_i}$ is used to normalize all the values into probabilities

$$\text{softmax}(z) = \left[\frac{\exp(z_1)}{\sum_{i=1}^{k} \exp(z_i)}, \frac{\exp(z_2)}{\sum_{i=1}^{k} \exp(z_i)}, \ldots, \frac{\exp(z_k)}{\sum_{i=1}^{k} \exp(z_i)} \right]$$

softmax: a generalization of sigmoid

- For a vector z of dimensionality k, the softmax is:

$$\text{softmax}(z) = \left[\frac{\exp(z_1)}{\sum_{i=1}^{k} \exp(z_i)}, \frac{\exp(z_2)}{\sum_{i=1}^{k} \exp(z_i)}, \ldots, \frac{\exp(z_k)}{\sum_{i=1}^{k} \exp(z_i)} \right]$$

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^{k} \exp(z_j)} \quad 1 \leq i \leq k$$

Example:

$$z = [0.6, 1.1, -1.5, 1.2, 3.2, -1.1]$$

$$\text{softmax}(z) = [0.055, 0.090, 0.006, 0.099, 0.74, 0.010]$$

Components of a probabilistic machine learning classifier

Given m input/output pairs $(x^{(i)}, y^{(i)})$:

1. A **feature representation** for the input. For each input observation $x^{(i)}$, a vector of features $[x_1, x_2, \ldots, x_n]$. Feature j for input $x^{(i)}$ is x_j, more completely $x_1^{(i)}$, or sometimes $f_j(x)$.

2. A **classification function** that computes $\hat{y}$ the estimated class, via $p(y|x)$, like the **sigmoid** or **softmax** functions

3. An **objective function** for learning, like **cross-entropy loss**

4. An algorithm for **optimizing** the objective function: **stochastic gradient descent**

Next class:

- Language models