Natural Language Processing

Text classification

Yulia Tsvetkov

yuliats@cs.washington.edu

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

Announcements

- HW1 is released! Kabir will review it today
- Quiz 1 is on Monday

Overview

- Three files:
  - Handout: Contains instructions, background on concepts we are using, and write up questions
  - Notebook A: Logistic Regression
  - Notebook B: N-Gram Language Models
- **Important**: If you downloaded files before the class, we recommend downloading them again as we have some minor changes
- The homework is divided into two modules, Module A is on Logistic Regression and Module B is on N-gram LMs.
- Each module has sub-parts and for each sub-part we have coding exercises and write-up questions
- We recommend getting started with the notebooks, and as you complete each coding exercises for a sub-part in a notebook, attempt the write up questions for that part in the handout

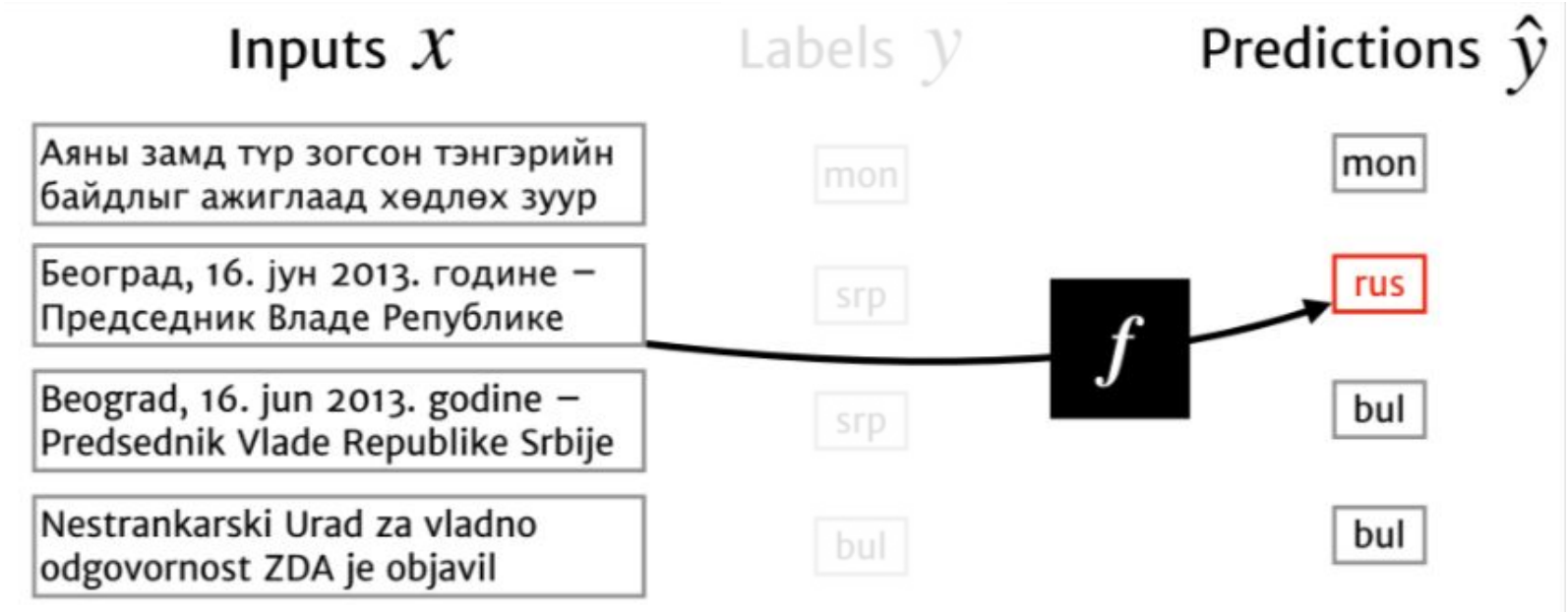
Overview: Using the Notebooks

- Recommended to run the notebooks on colab
- All the necessary background information is included in the notebooks as well (sometimes some equations might not properly render on colab, you can refer to that equation in the handout)
- For coding exercises you only need to implement the functions / classes. We provide all the function definitions similar to the Project 0
- Remove `raise NotImplementedError` statements when you implement your code.
- We provide sample test cases for the functions that you will be implementing, which you can use to check the correctness of your code.
- Once you finish implementing and testing all your code, **save the notebook as a .py file** and submit

Readings

- Eis 2 https://github.com/jacobeisenstein/gt-nlp-class/blob/master/notes/eisenstein-nlp-notes.pdf
- J&M III 4 https://web.stanford.edu/~jurafsky/slp3/4.pdf
- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. In Proceedings of EMNLP, 2002
- Andrew Y. Ng and Michael I. Jordan, On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes, In Proceedings of NeurIPS, 2001.

Text classification



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

Over the next couple of classes, we'll investigate:

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

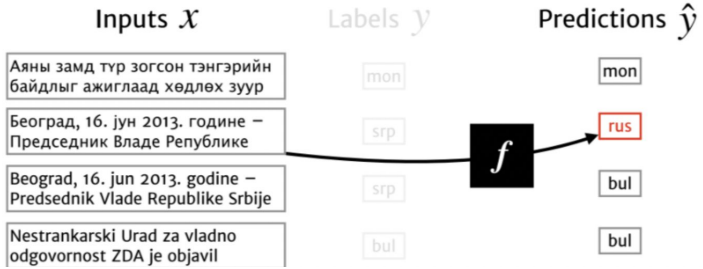
   (Keywords for this section: features, feature extraction, feature selection, representations)

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

   (Keyword for this section: models)

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

   (Keyword for this section: … evaluation)



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

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 ?

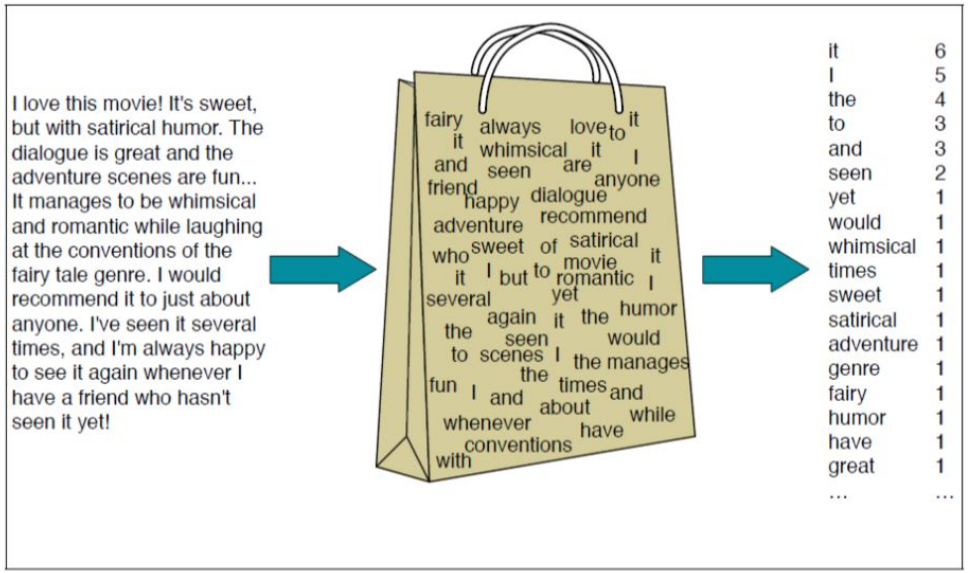


Figure from J&M 3rd ed. draft, sec 7.1

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
- 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

Over the next couple of classes, we'll investigate:

1. ~~How do we "digest" text into a form usable by a function?~~

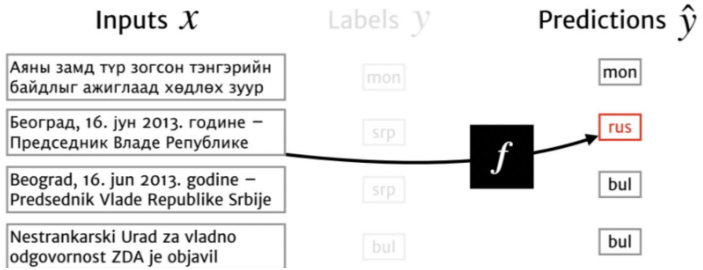
   ~~(Keywords for this section: features, feature extraction, feature selection, representations)~~

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

   (Keyword for this section: models)

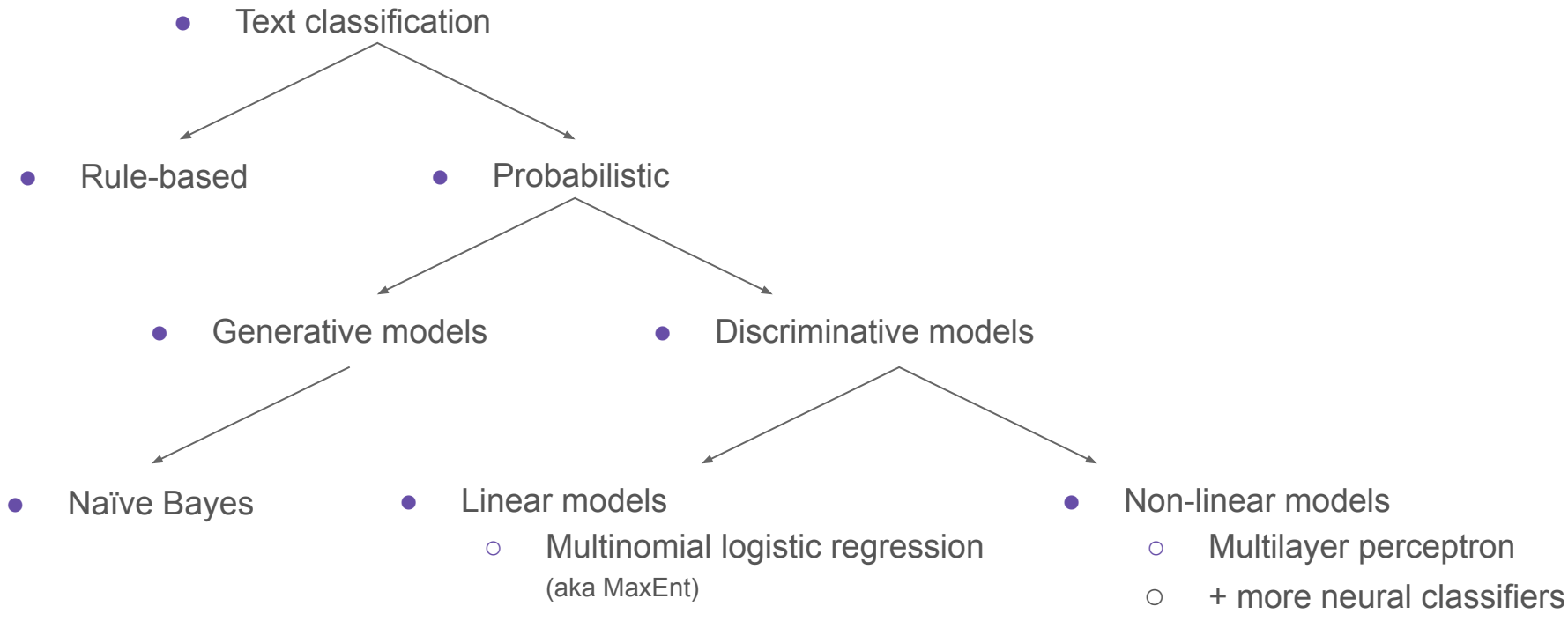
3. How do we evaluate our function *f*?

   (Keyword for this section: … evaluation)

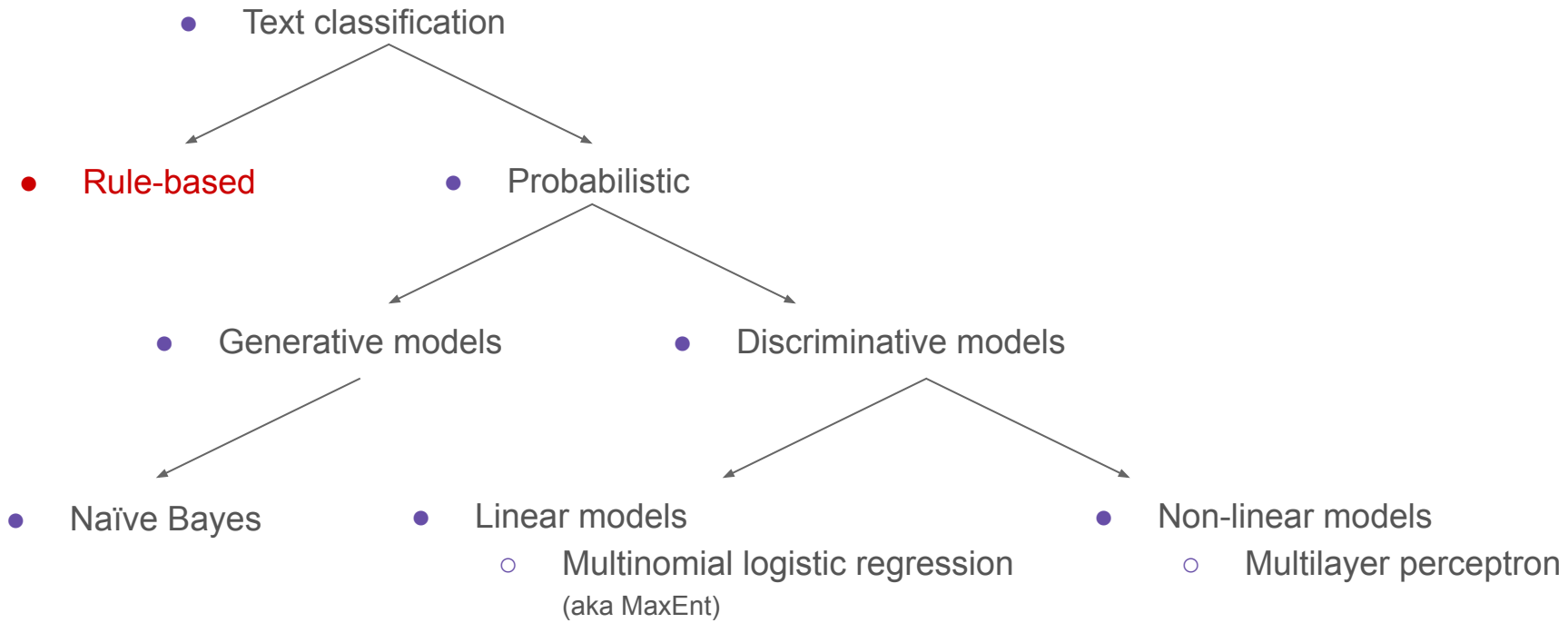


What kinds of strategies might we use to create our function *f*?

We'll consider alternative models for classification



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!

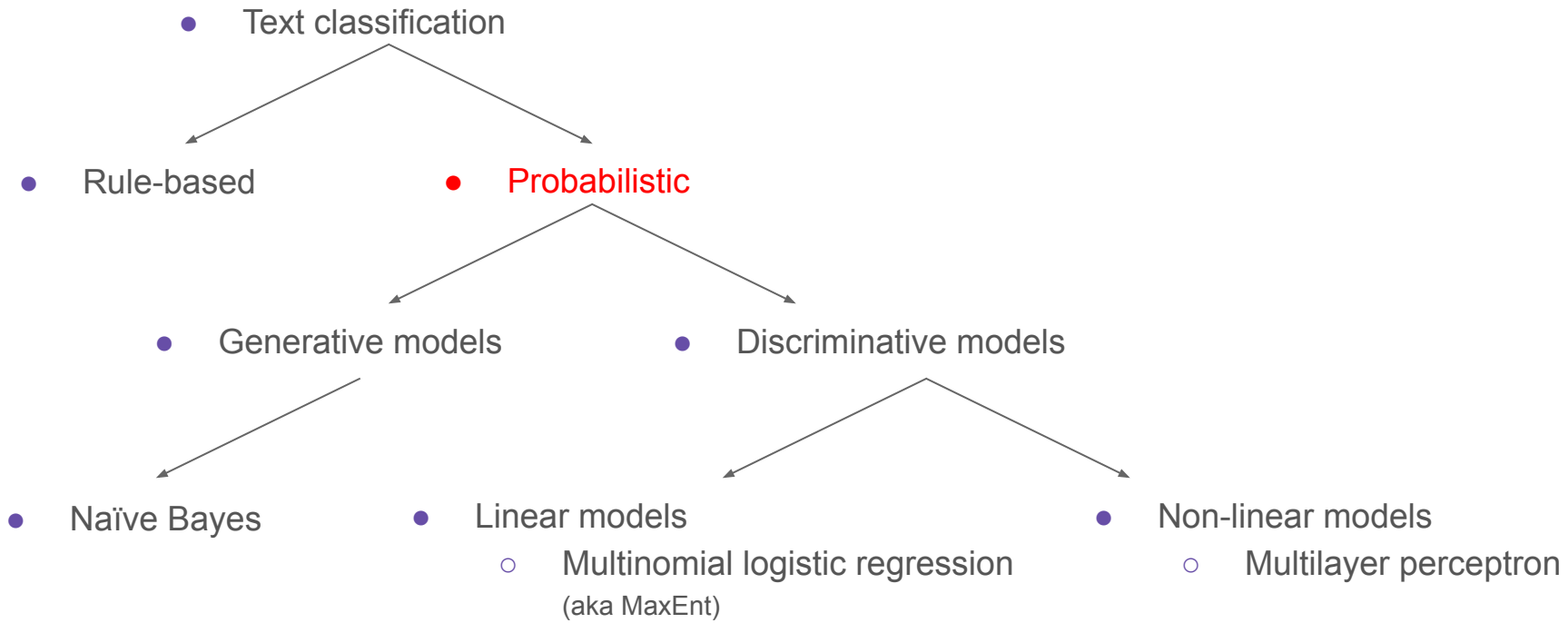
Language ID: All falter, stricken in kind.

⟶simple features can be misleading!

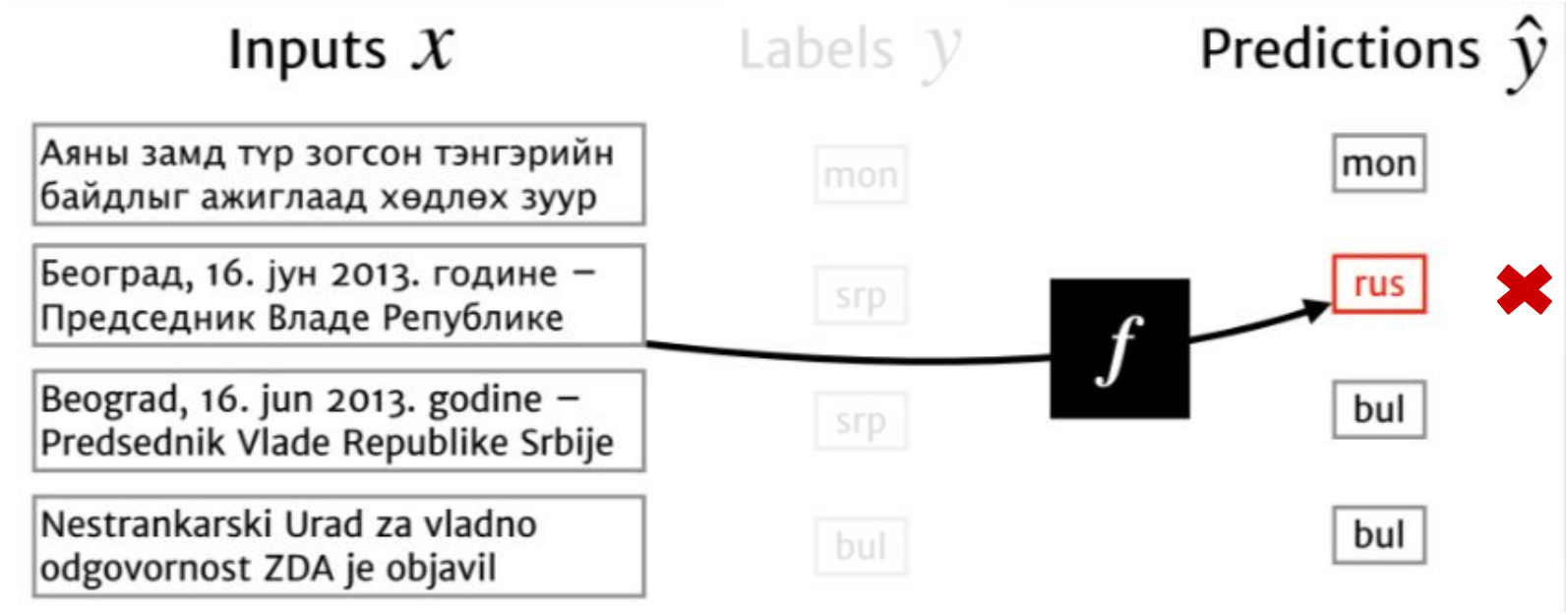
Rule-based classification

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~~

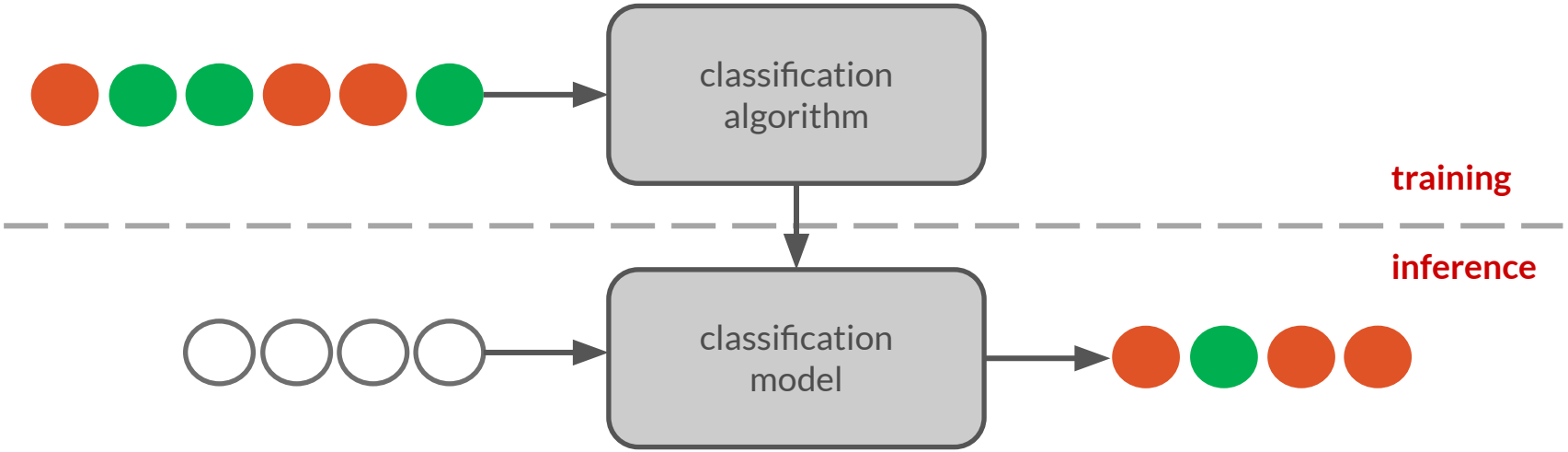
Classification: learning from data

- 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

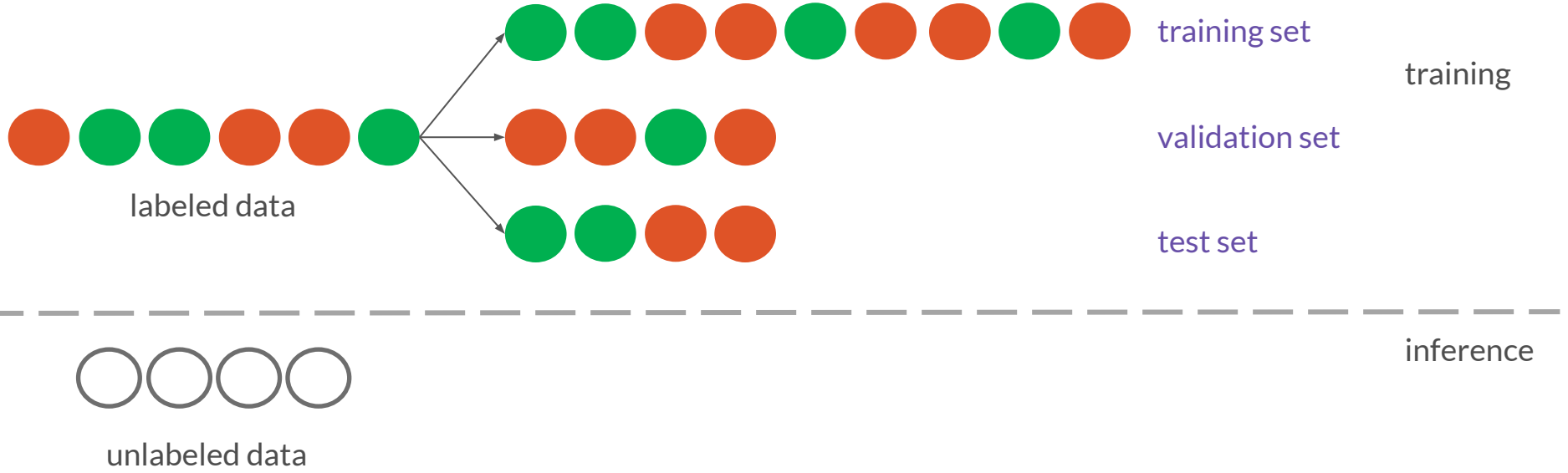
Where do datasets come from?

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

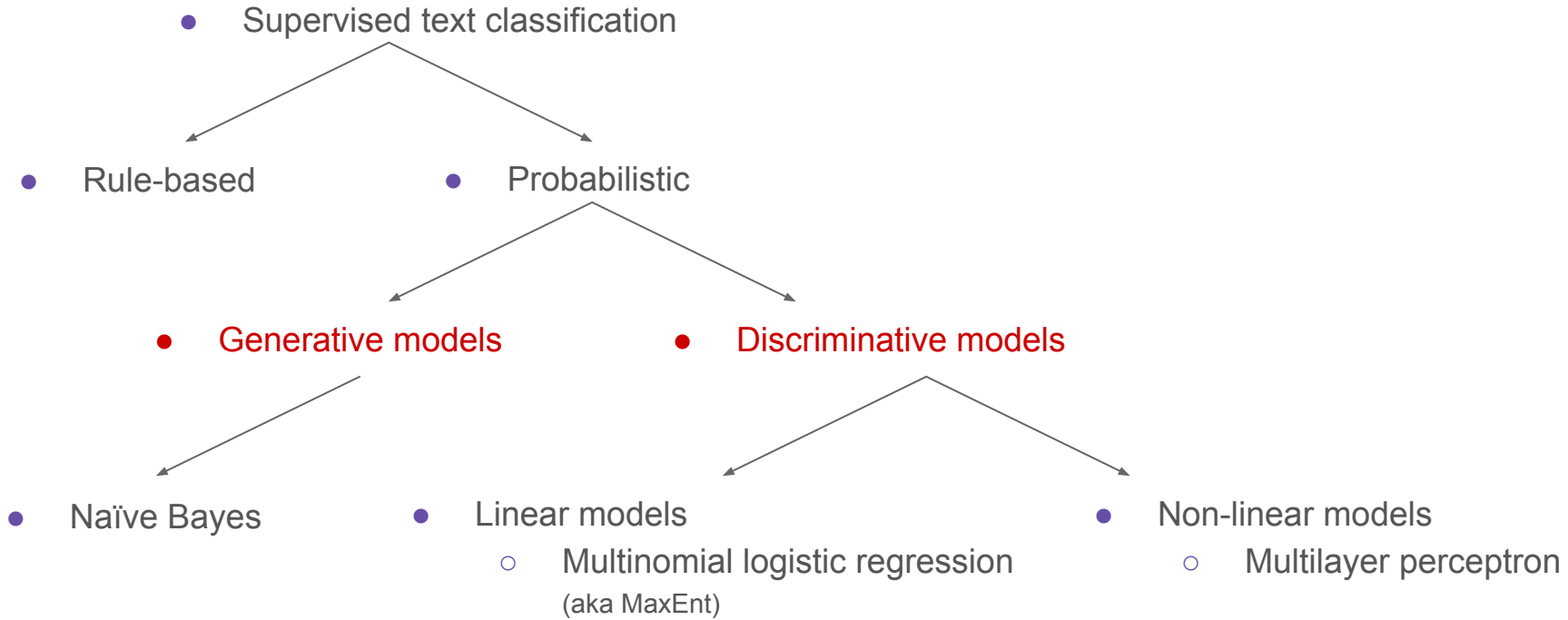
- Given data samples $\{x_1, x_2, \ldots, x_n\}$ and corresponding labels $Y = \{y_1, y_2, \ldots, y_k\}$
- We train a function $f: x \in X \rightarrow y \in Y$ (the model)

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

imagenet

imagenet

Now given a new image:

Run both models and see which one fits better

Discriminative model

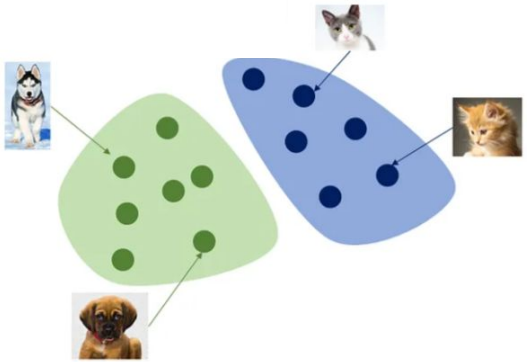
Just try to distinguish dogs from cats

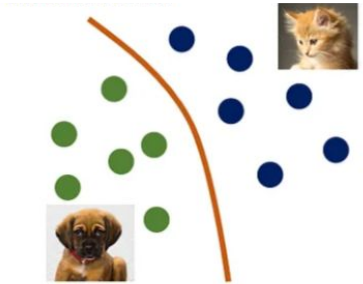
Oh look, dogs have collars! Let's ignore everything else

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

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

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

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

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

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

We'll consider alternative models for classification