Foundations of NLP

CS3126

Lecture-13

Probabilistic classifiers, Naïve Bayes

Mahindra™ University

Global Thinkers. Engaged Leaders.

Recap

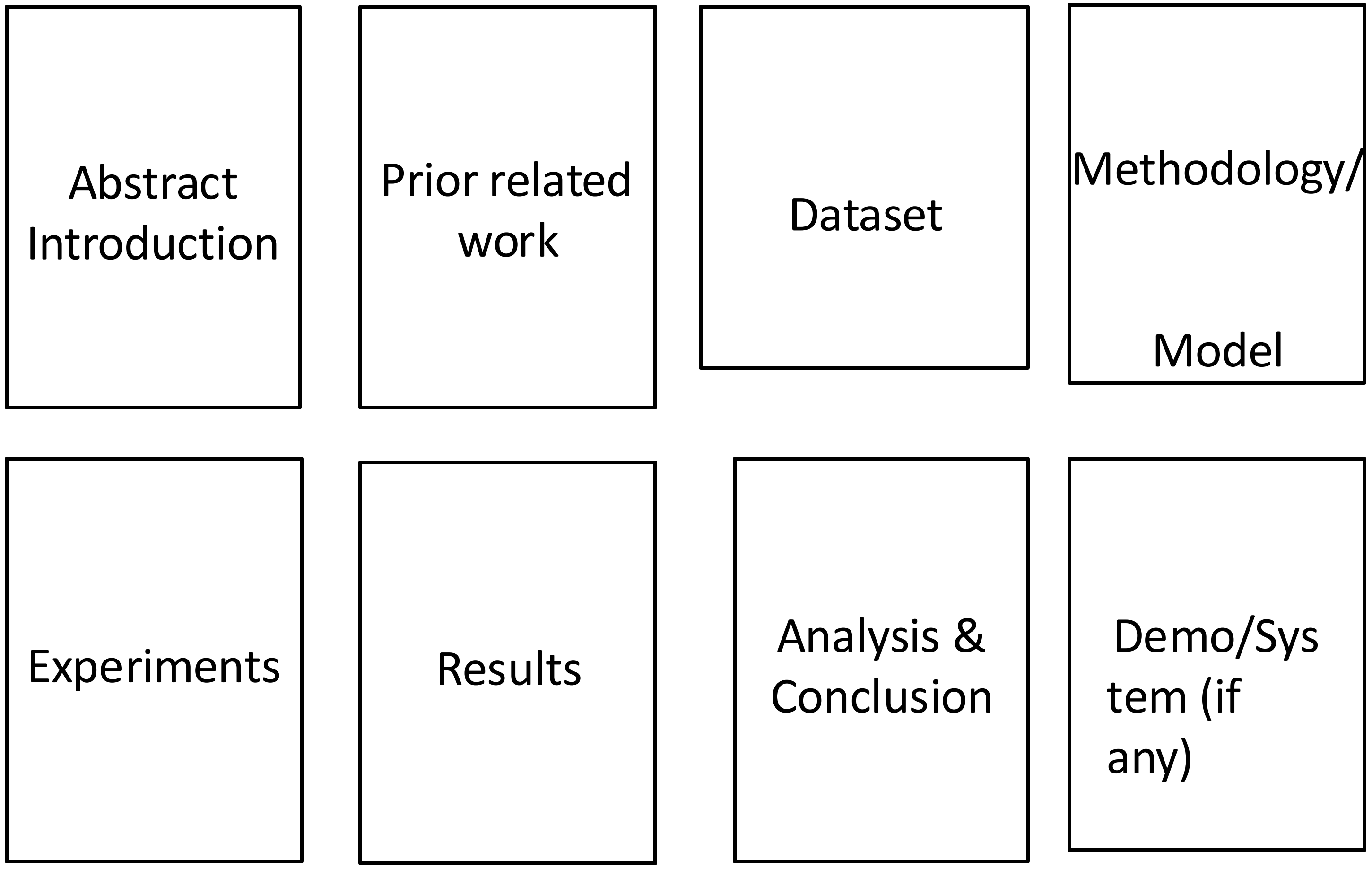
- NLP
- Applications
- Regular expressions
- Tokenization
- Stemming
    - Porter Stemmer
- Lemmatization
- Normalization
- Stopwords
- Bag-of-Words
- TF-IDF
- Probabilistic Classifiers
- **NER**
- **POS tagging**

Project Grading [Reminder Again]

- **Project Proposal refinement (5%)**: Improvement and refinement of the initial project proposal. Incorporation of feedback from the first evaluation phase.
- **Problem identification/Motivation/Dataset collection/Literature survey (7.5%)**: Critical analysis of existing work and identification of gaps. Highlight your innovation/uniqueness!
- **Methodology (7.5%)**- Appropriateness and rigor of the proposed methodology. clarity in explanation of methods, tools, and techniques. Justification of the chosen methods in relation to the problem.
- **Timeliness/Deliverables/Preliminary Final results (5%)**- Relevance and accuracy of data. Ability to analyze and interpret the results within the project's scope.
- **BONUS: Exceptional/Going Beyond - Research (3%)-** This bonus is awarded for those who take their projects a step further by translating their work into high-quality research or publication.
- **If you can demonstrate that your project has been developed into a well-researched paper or article suitable for a reputable venue, you will earn an additional 3% on your final grade.**

Final Project Report/Research project template
Deadline: 10 Dec 2024, 11:59 PM [A Form Link on Slack]

- Writeup quality is very important to your grade!!!
  - Look at recent years' papers/report for examples



Abstract Introduction

Prior related work

Dataset

Methodology/ Model

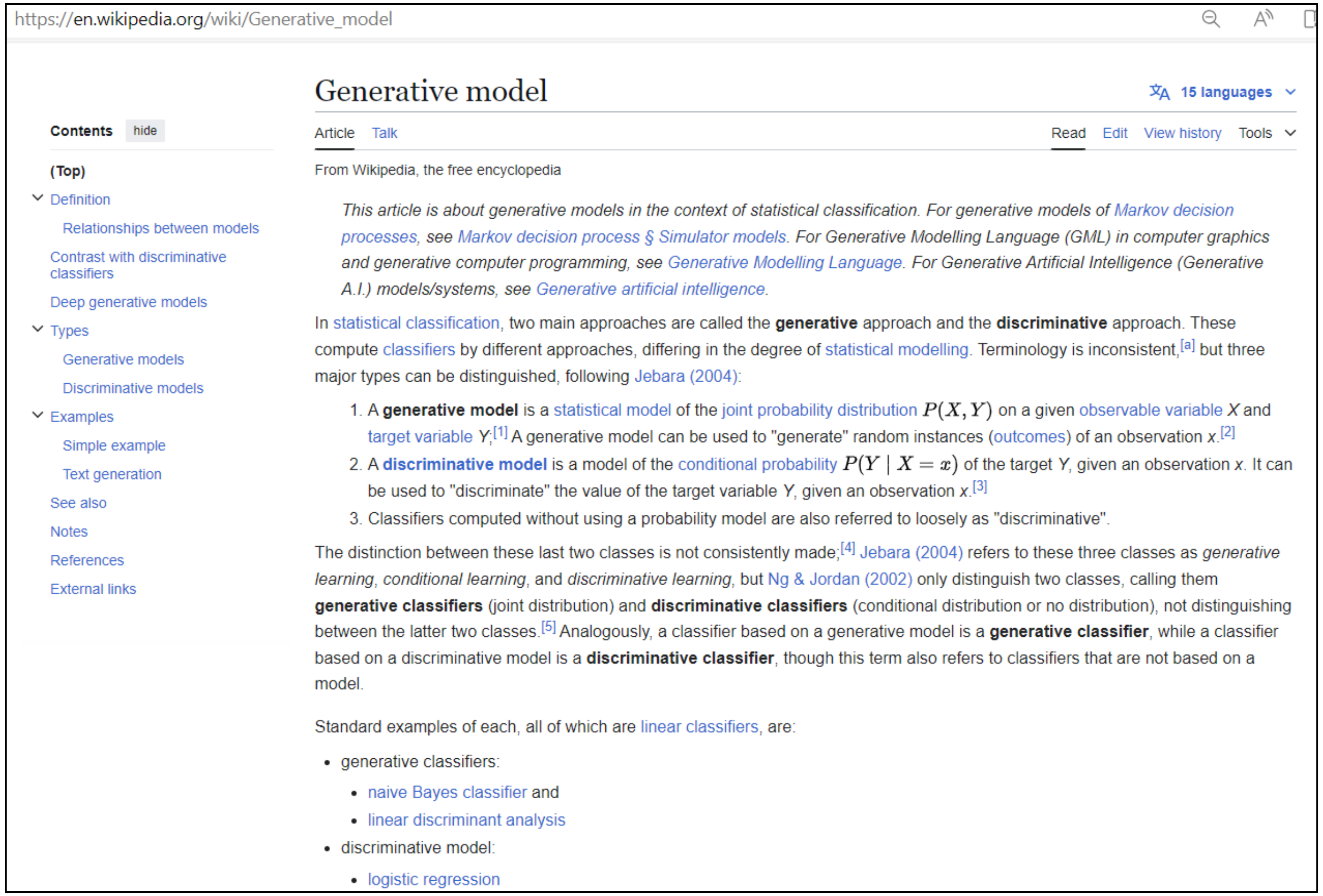
Experiments

Results

Analysis & Conclusion

Demo/System (if any)

What is a Generative Model?



Generative model

From Wikipedia, the free encyclopedia

This article is about generative models in the context of statistical classification. For generative models of Markov decision processes, see Markov decision process § Simulator models. For Generative Modelling Language (GML) in computer graphics and generative computer programming, see Generative Modelling Language. For Generative Artificial Intelligence (Generative A.I.) models/systems, see Generative artificial intelligence.

In statistical classification, two main approaches are called the **generative** approach and the **discriminative** approach. These compute classifiers by different approaches, differing in the degree of statistical modelling. Terminology is inconsistent,[a] but three major types can be distinguished, following Jebara (2004):

1. A **generative model** is a statistical model of the joint probability distribution $P(X,Y)$ on a given observable variable *X* and target variable *Y*;[1] A generative model can be used to "generate" random instances (outcomes) of an observation *x*.[2]
2. A **discriminative model** is a model of the conditional probability $P(Y \mid X=x)$ of the target *Y*, given an observation *x*. It can be used to "discriminate" the value of the target variable *Y*, given an observation *x*.[3]
3. Classifiers computed without using a probability model are also referred to loosely as "discriminative".

The distinction between these last two classes is not consistently made;[4] Jebara (2004) refers to these three classes as *generative learning*, *conditional learning*, and *discriminative learning*, but Ng & Jordan (2002) only distinguish two classes, calling them **generative classifiers** (joint distribution) and **discriminative classifiers** (conditional distribution or no distribution), not distinguishing between the latter two classes.[5] Analogously, a classifier based on a generative model is a **generative classifier**, while a classifier based on a discriminative model is a **discriminative classifier**, though this term also refers to classifiers that are not based on a model.

Standard examples of each, all of which are linear classifiers, are:

- generative classifiers:
  - naive Bayes classifier and
  - linear discriminant analysis
- discriminative model:
  - logistic regression

https://en.wikipedia.org/wiki/Generative_model

Consider yourself as an Instagram model

How Generative Modeling helps?

- Can generative modeling help generate apt possible unique combinations to choose for you?

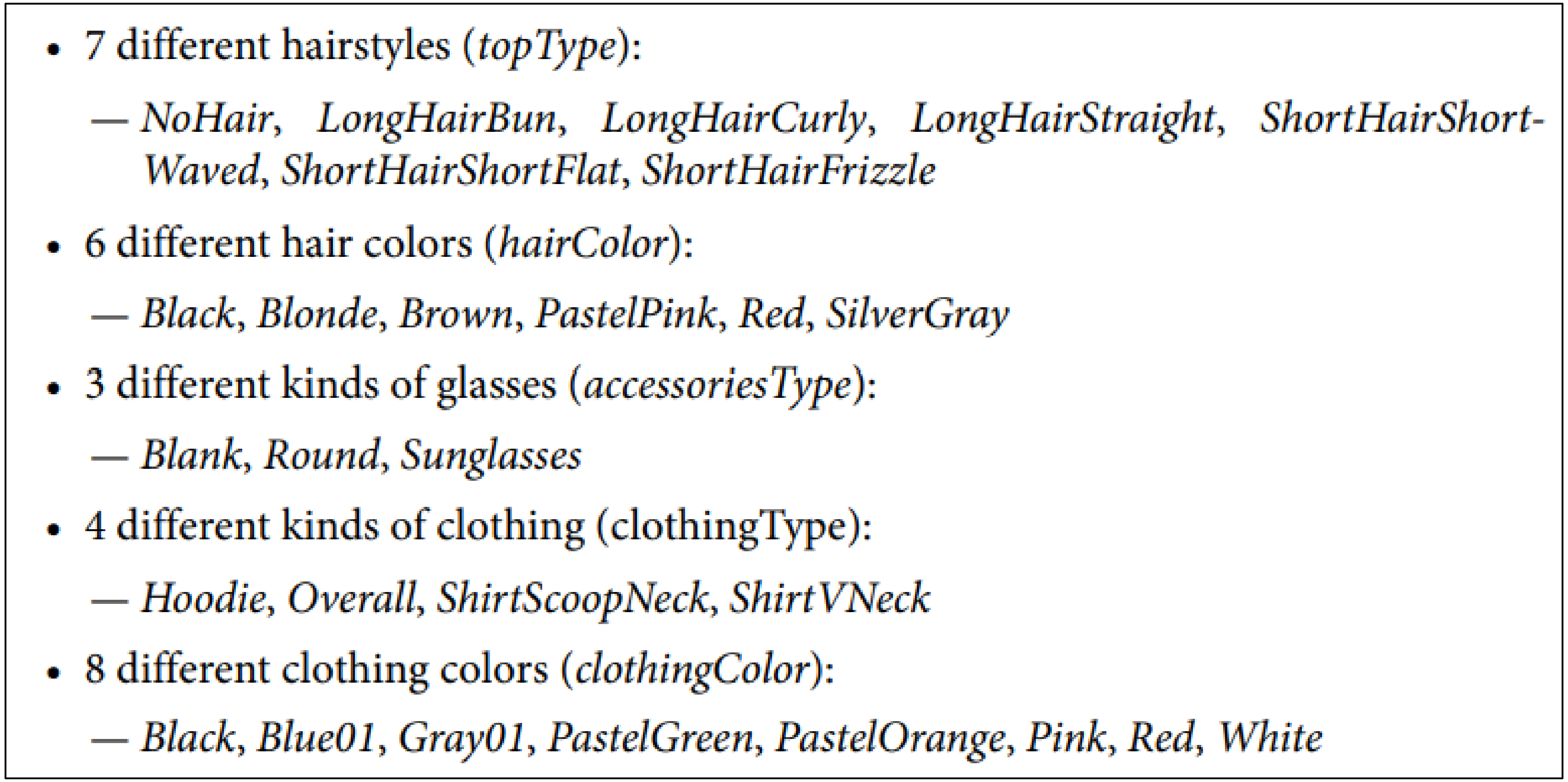
Features

You don’t have anything extra items other than those observed in the above images.

So, you can have just the given values for each of the ‘features’:

Consider yourself as an Instagram model

Possible Set of Features



- 7 different hairstyles (*topType*):
  - — *NoHair*, *LongHairBun*, *LongHairCurly*, *LongHairStraight*, *ShortHairShortWaved*, *ShortHairShortFlat*, *ShortHairFrizzle*
- 6 different hair colors (*hairColor*):
  - — *Black*, *Blonde*, *Brown*, *PastelPink*, *Red*, *SilverGray*
- 3 different kinds of glasses (*accessoriesType*):
  - — *Blank*, *Round*, *Sunglasses*
- 4 different kinds of clothing (clothingType):
  - — *Hoodie*, *Overall*, *ShirtScoopNeck*, *ShirtVNeck*
- 8 different clothing colors (*clothingColor*):
  - — *Black*, *Blue01*, *Gray01*, *PastelGreen*, *PastelOrange*, *Pink*, *Red*, *White*

https://medium.com/data-science-in-your-pocket/naive-bayes-as-a-generative-model-7fcc28787188

The entire sample space has **4032** points

Combinations

- **7(hairstyles)** x **6 (hair color)** x **3(specs)** x **4 (tops)** x **8 (colors)** =4032 unique combinations.

The entire sample space has **4032** points

- **Probability of last point** = 1-the summation of 4031 probabilities

How to calculate the probability ?

How to calculate the probability of these 4031 points (attire in our case)?

P(any attire) =

Frequency of an attire observed in-sample data/ Total samples

Red round Top with white hair & no glasses

- If you wore a
  - Red round Top with white hair & no glasses
  - 4 times in - 100 Instagram images

- P(Red & Round Top & white hair & No glasses)
  4/100

Drawback of such modeling (Probability)

- Such probability assignment assumes parameters to be almost 100% dependent on each other
  - They occur together (All)
  - They just don’t (Nothing)

$$P(A, B, C, D, E) = P(A \cap B \cap C \cap D \cap E)$$

Why? For Example, if you loved white hair, red round tops with glasses but you never thought of going without glasses with this combination

P (White hair, red round top without glasses) equivalent to P(Random hair with a random colored top with a random glass/no glass)

Due to absence of one feature

How about this?

P(A ,B ,C ,D , E) = P(A) x P(B) x P(C) x P(D) x P(E)

To avoid all-or-nothing,

If you get 4 /5 attributes you love, this combination has a higher chance of having a higher probability than a random combination!

Naïve Bayes Intitution

- Can generative modeling especially **Naive Bayes** help generate apt possible unique combinations to choose for you?

Why?

it assumes all the features are independent !!!

Choice of hairstyle has nothing to do with your top color, your top’s style is independent of the accessories you wear and so on!!!!!

How to calculate the probability now?

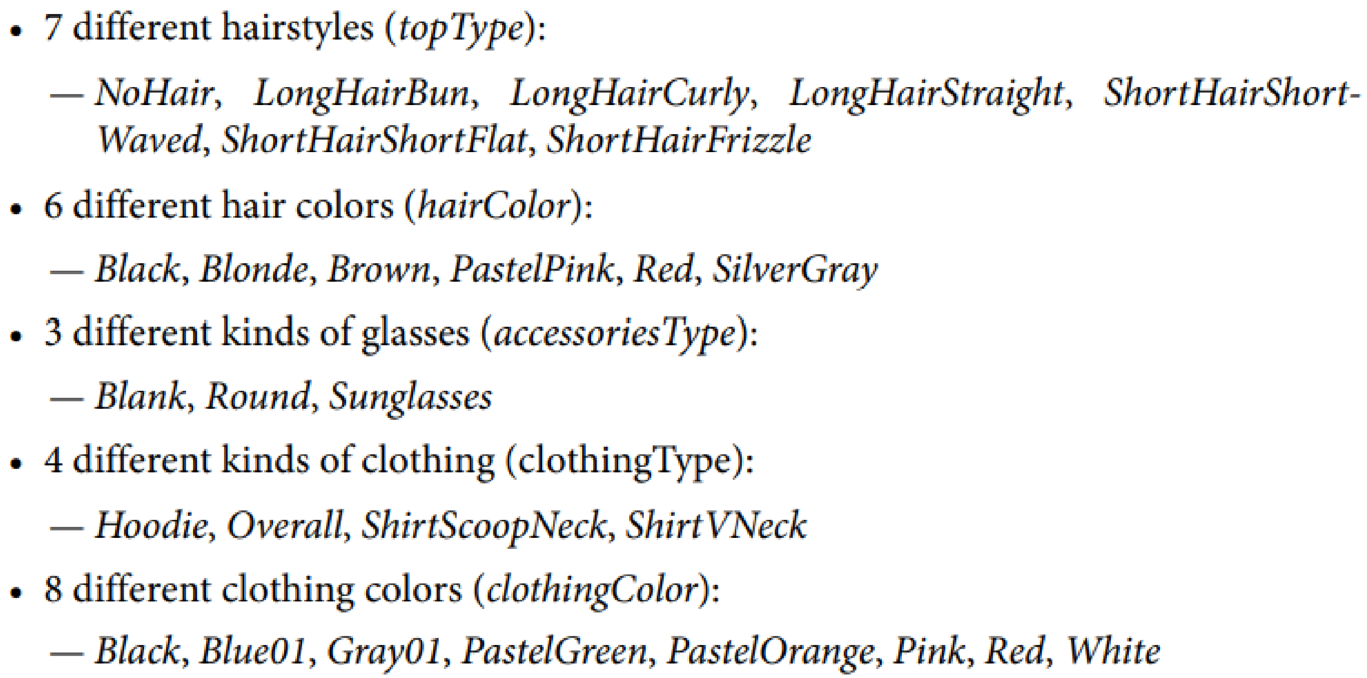
- ***P(Black hair, long curly hair, pink top, round top, no glasses) =***

P(Black hair) x P(long curly hair) x P(pink top) x P(round top) x P(no glasses)

where P(any element X) = occurrence of X / total samples

P(Black Hair) = *Frequency(Black hair)/Total Instagram images*

Reduced parameters? How?



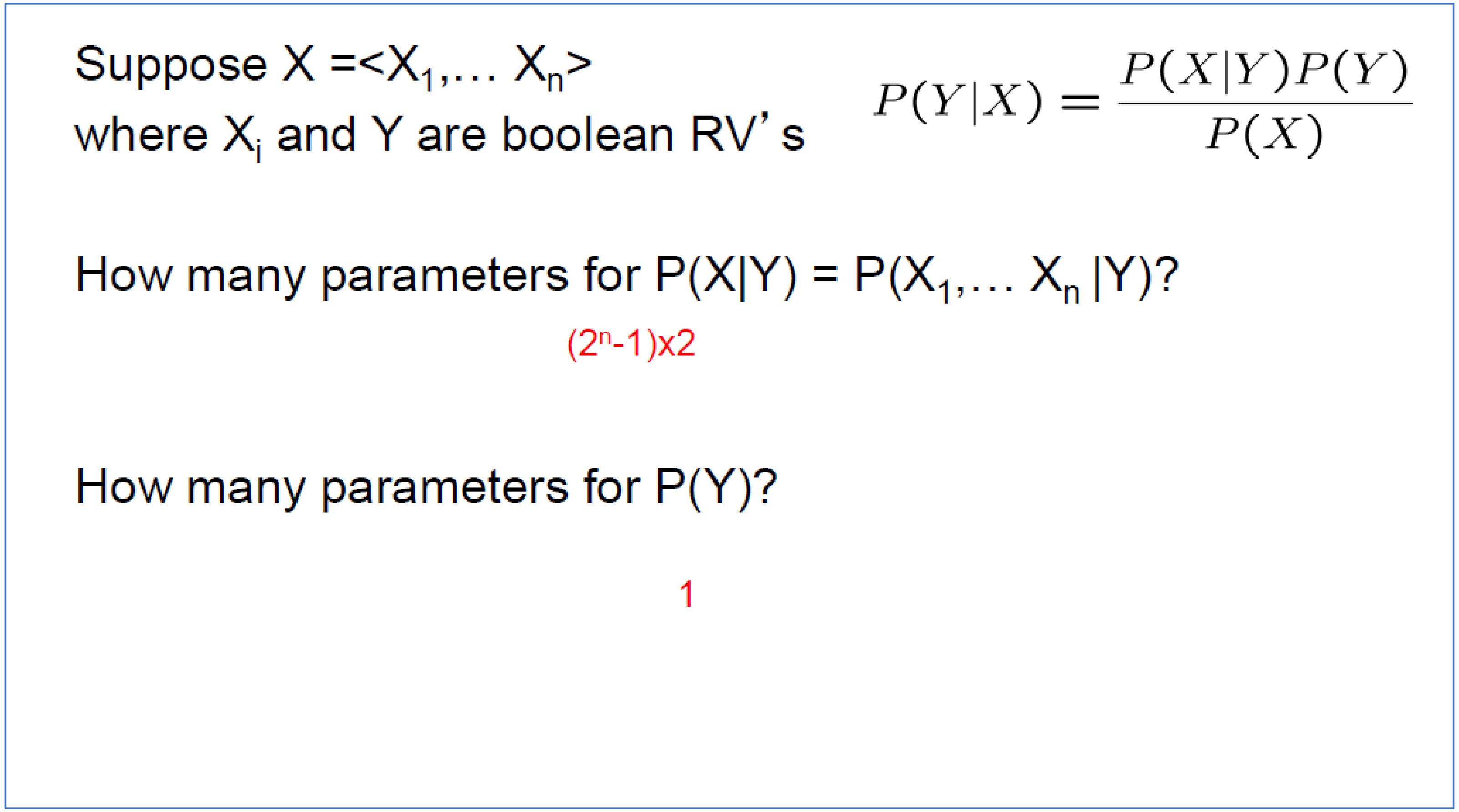
- 7 different hairstyles (*topType*):
  - — *NoHair*, *LongHairBun*, *LongHairCurly*, *LongHairStraight*, *ShortHairShortWaved*, *ShortHairShortFlat*, *ShortHairFrizzle*
- 6 different hair colors (*hairColor*):
  - — *Black*, *Blonde*, *Brown*, *PastelPink*, *Red*, *SilverGray*
- 3 different kinds of glasses (*accessoriesType*):
  - — *Blank*, *Round*, *Sunglasses*
- 4 different kinds of clothing (clothingType):
  - — *Hoodie*, *Overall*, *ShirtScoopNeck*, *ShirtVNeck*
- 8 different clothing colors (*clothingColor*):
  - — *Black*, *Blue01*, *Gray01*, *PastelGreen*, *PastelOrange*, *Pink*, *Red*, *White*

7(Hairstyle)–1 + 6(hair color)–1 + 3(glasses)-1 + 4(clothing type)-1 + 8(clothing color)-1 = 23

Reduced parameters? How?

- 4031 parameters in First case,
- Naive Bayes reduces this number to just 23 !!

Can we reduce the parameters using Bayes Rule?



Suppose X =<X_1,… X_n>
where X_i and Y are boolean RV’ s

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

How many parameters for P(X|Y) = P(X_1,… X_n |Y)?

(2^n-1)x2

How many parameters for P(Y)?

1

Back to Basics

- We have explored the probability of distribution of features to be learned!!

- This full distribution over the classes can be useful information for downstream decisions; avoiding making discrete decisions early on can be useful when combining systems

Probabilistic Classifier

- A probabilistic classifier additionally will tell us the probability of the observation being in the class.

- **Observation**
- **Learning**

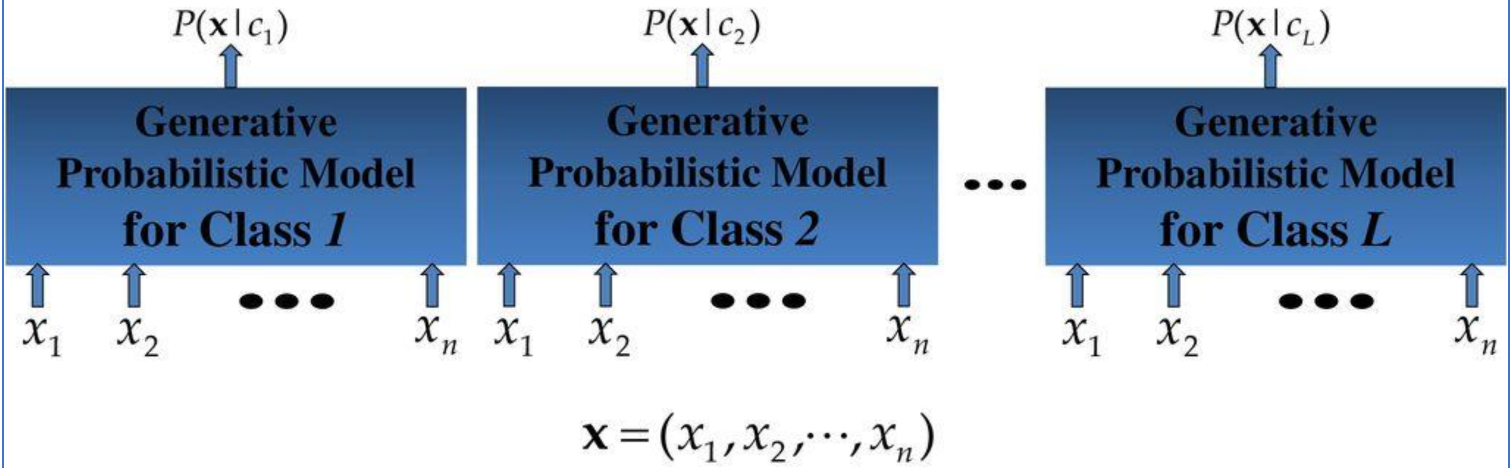
Probabilistic Classification

- Establishing a probabilistic model for classification (cont.)
  - **Generative model**

$$P(\mathbf{X}\,|\,C) \quad C = c_1, \cdots, c_L, \mathbf{X} = (X_1, \cdots, X_n)$$

Probability that this fruit is an apple

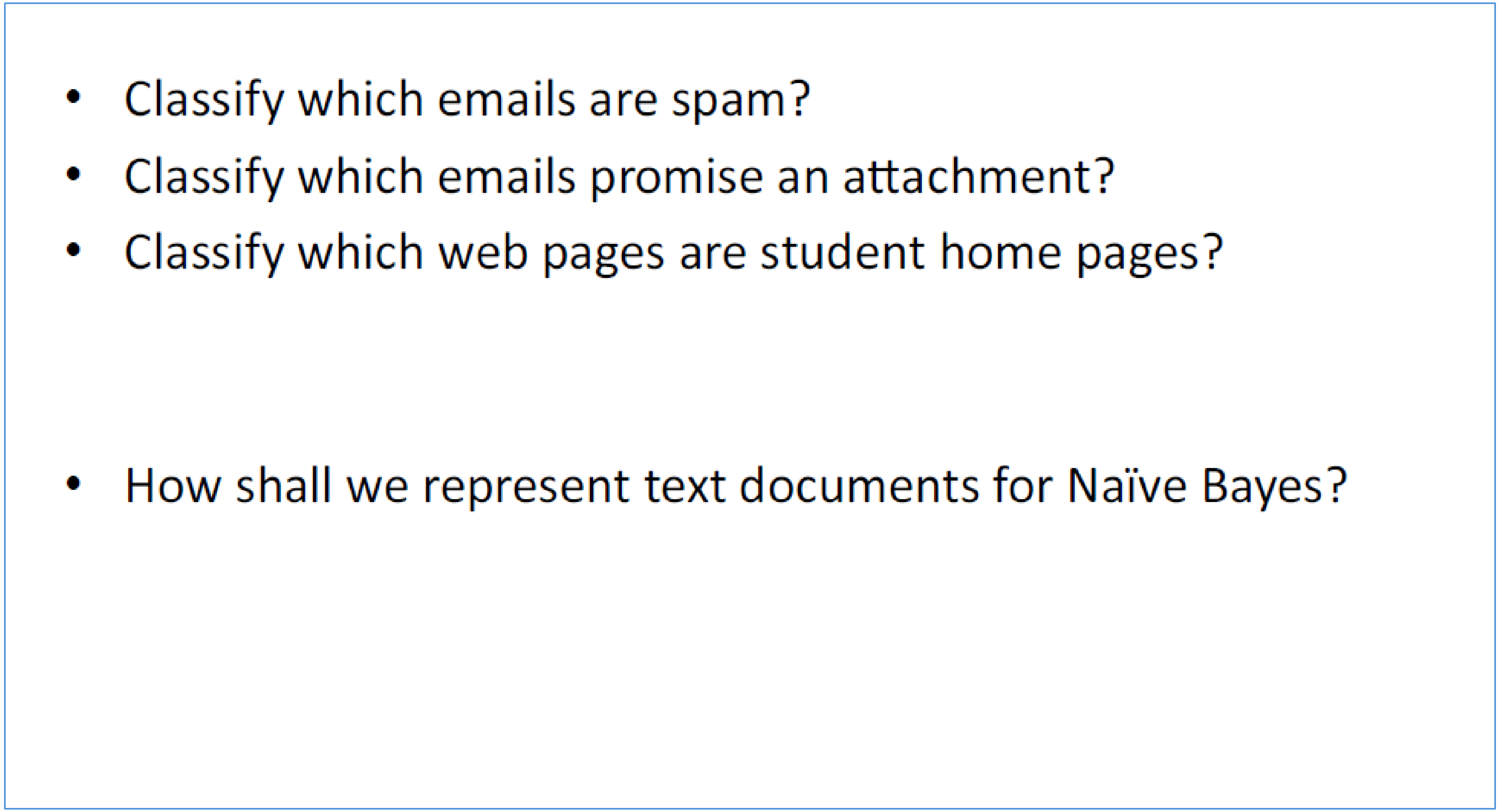
Probability that this fruit is an orange



$P(\mathbf{x}\,|\,c_1)$
Generative Probabilistic Model for Class *1*
x_1 x_2 ••• x_n
$P(\mathbf{x}\,|\,c_2)$
Generative Probabilistic Model for Class *2*
x_1 x_2 ••• x_n
•••
$P(\mathbf{x}\,|\,c_L)$
Generative Probabilistic Model for Class *L*
x_1 x_2 ••• x_n
$\mathbf{x} = (x_1, x_2, \cdots, x_n)$

Vectors of random variables

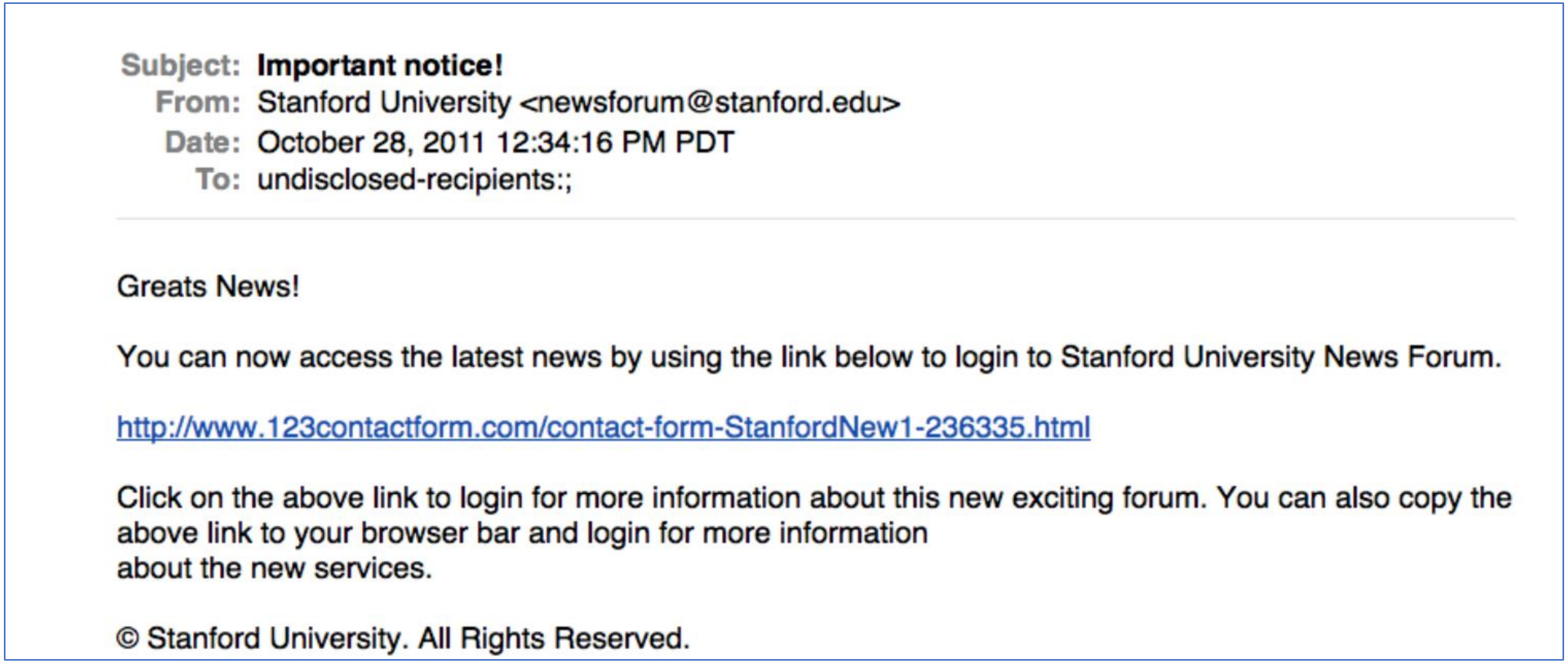
Learning to classify text documents



- Classify which emails are spam?
- Classify which emails promise an attachment?
- Classify which web pages are student home pages?

- How shall we represent text documents for Naïve Bayes?

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.

https://web.stanford.edu/class/cs124/lec/naivebayes2021.pdf

Text Classification

- Assigning subject categories, topics, or genres
- Spam detection
- Authorship identification
- Age/gender identification
- Language Identification
- Sentiment analysis
- …

Text Classification: definition

- *Input*:
  - a document d
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$

- *Output*: a predicted class $c \in C$

Bayes’ Rule Applied to Documents and Classes

- For a document *d* and a class *c*

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

Naïve Bayes Classifier (I)

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(c \mid d)$$

MAP is "maximum a posteriori" = most likely class

$$= \underset{c \in C}{\operatorname{argmax}} \frac{P(d \mid c) P(c)}{P(d)}$$

Bayes Rule

$$= \underset{c \in C}{\operatorname{argmax}} P(d \mid c) P(c)$$

Dropping the denominator

Naïve Bayes Classifier (II)

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} \, P(d \mid c) P(c)$$

$$= \underset{c \in C}{\operatorname{argmax}} \, P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

Document d represented as features x1..xn

Naïve Bayes Classifier (IV)

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

O(|X|^n•|C|) parameters

Could only be estimated if a very, very large number of training examples was available.

How often does this class occur?

We can just count the relative frequencies in a corpus

Multinomial Naïve Bayes Independence Assumptions

$$P(x_1, x_2, \ldots, x_n \mid c)$$

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

$$P(x_1, \ldots, x_n \mid c) = P(x_1 \mid c) \bullet P(x_2 \mid c) \bullet P(x_3 \mid c) \bullet \ldots \bullet P(x_n \mid c)$$

Learning the Multinomial Naïve Bayes Model

- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{doccount(C = c_j)}{N_{doc}}$$

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

Parameter estimation

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

fraction of times word w_i appears among all words in documents of topic c_j

- Create mega-document for topic *j* by concatenating all docs in this topic
  - Use frequency of *w* in mega-document

Laplace (add-1) smoothing for Naïve Bayes

$$\hat{P}(w_i \mid c) = \frac{count(w_i,c)+1}{\sum_{w\in V}\left(count(w,c))+1\right)}$$

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

Multinomial Naïve Bayes: Learning

- From training corpus, extract *Vocabulary*

- Calculate $P(c_j)$ terms
  - For each c_j in C do

    $docs_j \leftarrow$ all docs with class $=c_j$

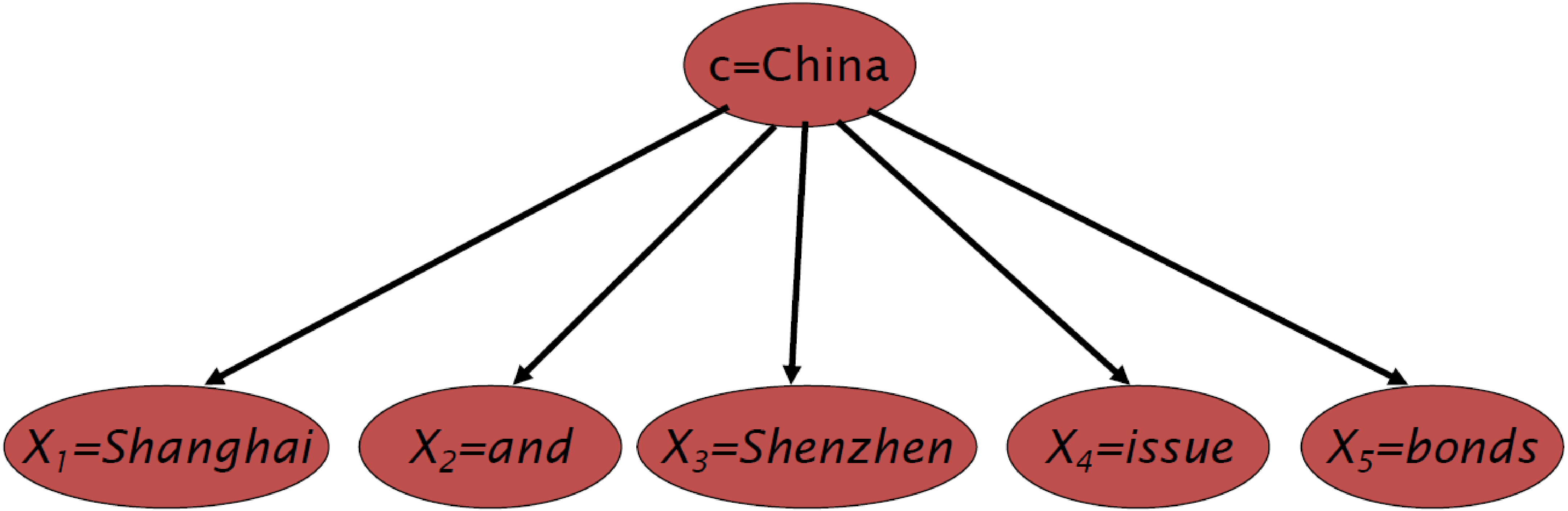
    $$P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$$

- Calculate $P(w_k \mid c_j)$ terms
  - $Text_j \leftarrow$ single doc containing all $docs_j$
  - Foreach word w_k in *Vocabulary*

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

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

Generative Model for Multinomial Naïve Bayes



$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

Priors:

$P(c) = \frac{3}{4}$

$P(j) = \frac{1}{4}$

Conditional Probabilities:

P(Chinese|*c*) = (5+1) / (8+6) = 6/14 = 3/7

P(Tokyo|*c*) = (0+1) / (8+6) = 1/14

P(Japan|*c*) = (0+1) / (8+6) = 1/14

P(Chinese|*j*) = (1+1) / (3+6) = 2/9

P(Tokyo|*j*) = (1+1) / (3+6) = 2/9

P(Japan|*j*) = (1+1) / (3+6) = 2/9

Choosing a class:

$P(c|d5) \propto 3/4 * (3/7)^3 * 1/14 * 1/14 \approx 0.0003$

$P(j|d5) \propto 1/4 * (2/9)^3 * 2/9 * 2/9 \approx 0.0001$

$$P\left(\frac{c}{d_5}\right) = \frac{P(c)\,P\left(\frac{d_5}{c}\right)}{P(d_5)}$$

Bayes Rule

Test document

d_5 → Chinese Chinese Chinese Tokyo Japan

$P(c) = 3/4$

$P(j) = 1/4$

$$\frac{3}{4} \times P\left(\frac{d_5}{c}\right)$$

$$P\left(\frac{d_5}{c}\right) = P\left(\frac{\text{Chinese Chinese Chinese Tokyo Japan}}{c = \text{China}}\right)$$

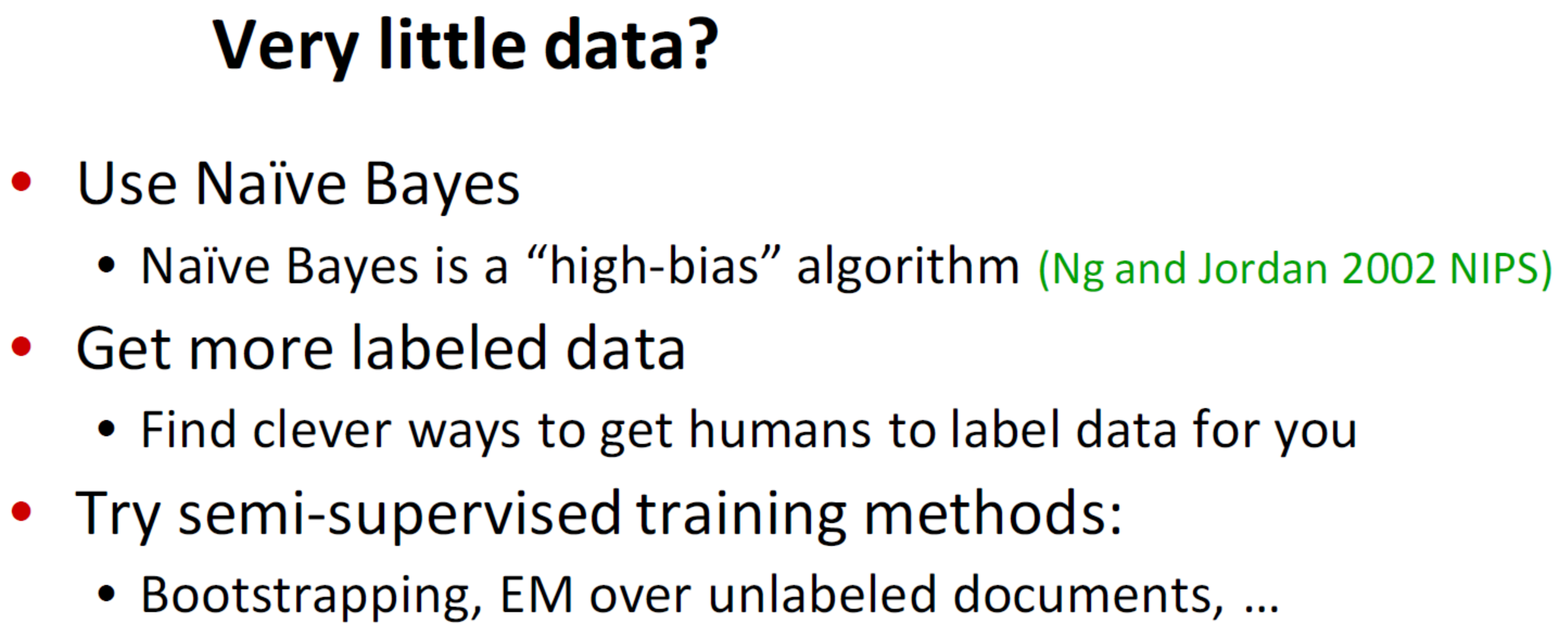
$$= P\left(\frac{\text{Chinese}}{\cancel{\text{chi}}\ \text{China}}\right) P\left(\frac{\text{Chinese}}{\text{China}}\right) \times P\left(\frac{\text{Chinese}}{\text{China}}\right) \times P\left(\frac{\text{Tokyo}}{\text{China}}\right) \times P\left(\frac{\text{Japan}}{\text{China}}\right)$$

$$\frac{3}{7} \times \frac{3}{7} \times \frac{3}{7} \times \frac{1}{14} \times \frac{1}{14}$$

$$\boxed{\frac{3}{4}\left(\frac{3}{7}\right)^3 \times \left(\frac{1}{14}\right)^2}$$

denominator is same for both $P\left(\frac{c}{d_5}\right)$ & $P\left(\frac{j}{d_5}\right)$

Limitations



Very little data?

- Use Naïve Bayes
  - Naïve Bayes is a “high-bias” algorithm (Ng and Jordan 2002 NIPS)
- Get more labeled data
  - Find clever ways to get humans to label data for you
- Try semi-supervised training methods:
  - Bootstrapping, EM over unlabeled documents, …

Acknowledgments

- These slides were adapted from the book SPEECH and LANGUAGE PROCESSING: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition
- Many of these slides are derived from Stanford Text Classification, Tom Mitchell, William Cohen, Eric Xing and Seyoung Kim. Thanks!
- References: https://web.stanford.edu/class/cs124/lec/naivebayes2021.pdf