Naïve Bayes & Logistic Regression
Lecture 18

David Sontag

New York University

Slides adapted from Vibhav Gogate, Luke Zettlemoyer, Carlos Guestrin, and Dan Weld

Naïve Bayes

- Naïve Bayes assumption:
  - Features are independent given class:

$$P(X_1, X_2|Y) = P(X_1|X_2, Y)P(X_2|Y)$$
$$= P(X_1|Y)P(X_2|Y)$$

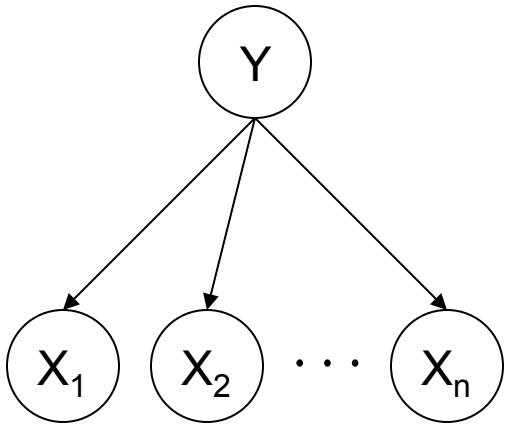
  - More generally:

$$P(X_1 \ldots X_n|Y) = \prod_i P(X_i|Y)$$

- How many parameters now?
  - Suppose **X** is composed of *n* binary features

The Naïve Bayes Classifier

- Given:
  - Prior P(Y)
  - *n* conditionally independent features **X** given the class Y
  - For each X_i, we have likelihood $P(X_i|Y)$



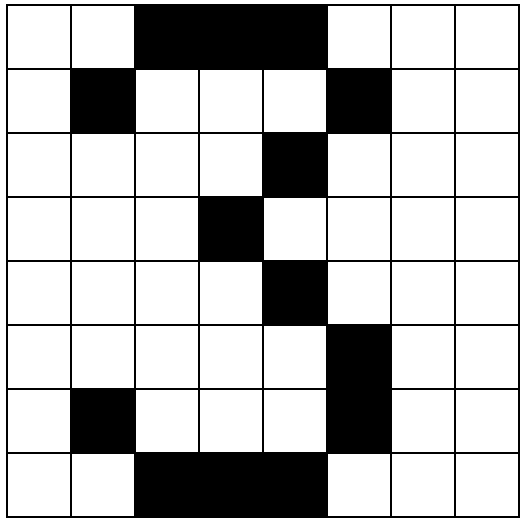
- Decision rule:

$$y^* = h_{NB}(\mathbf{x}) = \arg\max_y P(y) P(x_1, \ldots, x_n \mid y)$$
$$= \arg\max_y P(y) \prod_i P(x_i|y)$$

If certain assumption holds, NB is optimal classifier!
(they typically don't)

A Digit Recognizer

- Input: pixel grids



- Output: a digit 0-9

Naïve Bayes for Digits (Binary Inputs)

- Simple version:
  - One feature F_{ij} for each grid position <i,j>
  - Possible feature values are on / off, based on whether intensity is more or less than 0.5 in underlying image
  - Each input maps to a feature vector, e.g.

$$\rightarrow \langle F_{0,0} = 0 \;\; F_{0,1} = 0 \;\; F_{0,2} = 1 \;\; F_{0,3} = 1 \;\; F_{0,4} = 0 \;\; \ldots F_{15,15} = 0 \rangle$$

  - Here: lots of features, each is binary valued
- Naïve Bayes model:

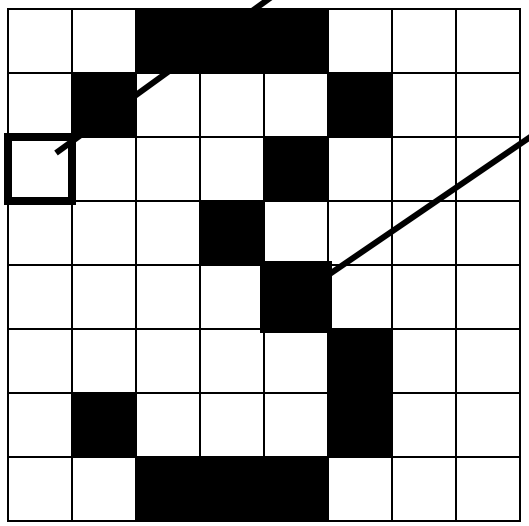
$$P(Y|F_{0,0} \ldots F_{15,15}) \propto P(Y) \prod_{i,j} P(F_{i,j}|Y)$$

- Are the features independent given class?
- What do we need to learn?

Example Distributions

$P(Y)$

1	0.1
2	0.1
3	0.1
4	0.1
5	0.1
6	0.1
7	0.1
8	0.1
9	0.1
0	0.1



$P(F_{3,1} = on|Y)$

1	0.01
2	0.05
3	0.05
4	0.30
5	0.80
6	0.90
7	0.05
8	0.60
9	0.50
0	0.80

$P(F_{5,5} = on|Y)$

1	0.05
2	0.01
3	0.90
4	0.80
5	0.90
6	0.90
7	0.25
8	0.85
9	0.60
0	0.80

MLE for the parameters of NB

- Given dataset
  - Count(A=a,B=b) $\leftarrow$ number of examples where A=a and B=b
- MLE for discrete NB, simply:
  - Prior:

$$P(Y = y) = \frac{Count(Y = y)}{\sum_{y'} Count(Y = y')}$$

  - Likelihood:

$$P(X_i = x|Y = y) = \frac{Count(X_i = x, Y = y)}{\sum_{x'} Count(X_i = x', Y = y)}$$

Subtleties of NB classifier – Violating the NB assumption

- Usually, features are not conditionally independent:

$$P(X_1 \ldots X_n | Y) \neq \prod_i P(X_i | Y)$$

- Actual probabilities P(Y|**X**) often biased towards 0 or 1 (i.e., not well "calibrated")

- Nonetheless, NB often performs well, even when assumption is violated

Text classification

- Classify e-mails
  - Y = {Spam,NotSpam}
- Classify news articles
  - Y = {what is the topic of the article?}
- Classify webpages
  - Y = {Student, professor, project, …}

- What about the features **X**?
  - The text!

Features **X** are entire document – X_i for i^{th} word in article

Article from rec.sport.hockey

```
Path: cantaloupe.srv.cs.cmu.edu!das-news.harvard.e
From: xxx@yyy.zzz.edu (John Doe)
Subject: Re: This year's biggest and worst (opinic
Date: 5 Apr 93 09:53:39 GMT

I can only comment on the Kings, but the most
obvious candidate for pleasant surprise is Alex
Zhitnik. He came highly touted as a defensive
defenseman, but he's clearly much more than that.
Great skater and hard shot (though wish he were
more accurate). In fact, he pretty much allowed
the Kings to trade away that huge defensive
liability Paul Coffey. Kelly Hrudey is only the
biggest disappointment if you thought he was any
good to begin with. But, at best, he's only a
mediocre goaltender. A better choice would be
Tomas Sandstrom, though not through any fault of
his own, but because some thugs in Toronto decided
```

Bag of Words Approach

aardvark	0
about	2
all	2
Africa	1
apple	0
anxious	0
...	
gas	1
...	
oil	1
…	
Zaire	0

NB with Bag of Words for text classification

- Learning phase:
  - Prior P(Y)
    - Count number of documents for each topic
  - $P(X_i|Y)$
    - Just considering the documents assigned to topic Y, find the fraction of docs containing a given word; remember this dist'n is shared across all positions i
- Test phase:
  - For each document
    - Use naïve Bayes decision rule

$$h_{NB}(\mathbf{x}) = \arg\max_y P(y) \prod_{i=1}^{LengthDoc} P(x_i|y)$$

Naive Bayes = Linear Classifier

- Theorem: assume that $x_i \in \{0, 1\}$ for all $i \in [1, N]$. Then, the Naive Bayes classifier is defined by

$$\mathbf{x} \mapsto \operatorname{sgn}(\mathbf{w} \cdot \mathbf{x} + b),$$

[Slide from Mehyrar Mohri]

Summary

- Bayesian prediction:
  - requires solving density estimation problems.
  - often difficult to estimate $\Pr[\mathbf{x} \mid y]$ for $\mathbf{x} \in \mathbb{R}^N$.
  - but, simple and easy to apply; widely used.
- Naive Bayes:
  - strong assumption.
  - straightforward estimation problem.
  - specific linear classifier.
  - sometimes surprisingly good performance.

Lets take a(nother) probabilistic approach!!!

- Previously: directly estimate the data distribution P(X,Y)!
  - challenging due to size of distribution!
  - make Naïve Bayes assumption: only need $P(X_i|Y)$!
- But wait, we classify according to:
  - $\max_Y P(Y|X)$
- Why not learn P(Y|X) directly?

mpg	cylinders	displacemen	horsepower	weight	acceleration	modelyear	maker
good	4	97	75	2265	18.2	77	asia
bad	6	199	90	2648	15	70	america
bad	4	121	110	2600	12.8	77	europe
bad	8	350	175	4100	13	73	america
bad	6	198	95	3102	16.5	74	america
bad	4	108	94	2379	16.5	73	asia
bad	4	113	95	2228	14	71	asia
bad	8	302	139	3570	12.8	78	america
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
good	4	120	79	2625	18.6	82	america
bad	8	455	225	4425	10	70	america
good	4	107	86	2464	15.5	76	europe
bad	5	131	103	2830	15.9	78	europe

Generative vs. Discriminative Classifiers

- **Want to Learn**: $\mathbf{X} \mapsto Y$
  - **X** – features
  - Y – target classes

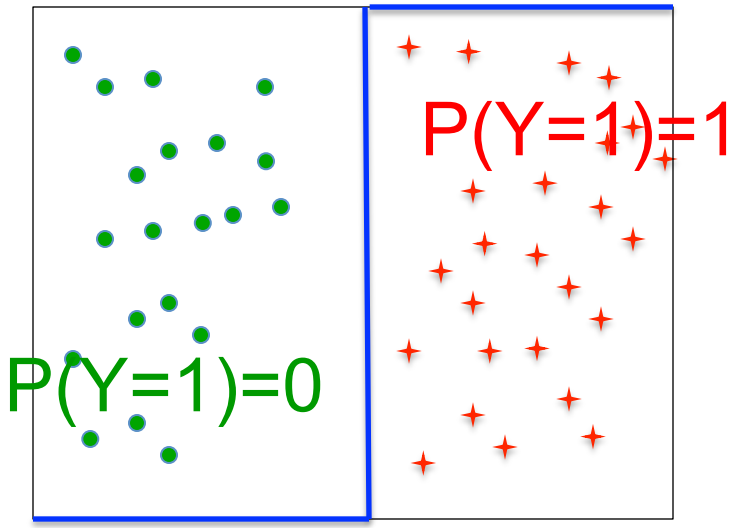
$$P(Y \mid \mathbf{X}) \propto P(\mathbf{X} \mid Y)\, P(Y)$$

- **Generative classifier**, e.g., Naïve Bayes:
  - Assume some **functional form for P(X|Y), P(Y)**
  - Estimate parameters of P(X|Y), P(Y) directly from training data
  - Use Bayes' rule to calculate P(Y|X= x)
  - This is an example of a **"*generative*" model**
    - **Indirect** computation of P(Y|X) through Bayes rule
    - As a result, **can also generate a sample of the data**, $P(X) = \sum_y P(y)\, P(X|y)$
  - **Can easily handle missing data**

- **Discriminative classifiers**, e.g., Logistic Regression:
  - Assume some **functional form for P(Y|X)**
  - Estimate parameters of P(Y|X) directly from training data
  - This is the **"*discriminative*" (or "conditional") model**
    - Directly learn P(Y|X)
    - But **cannot obtain a sample of the data**, because P(X) is not available

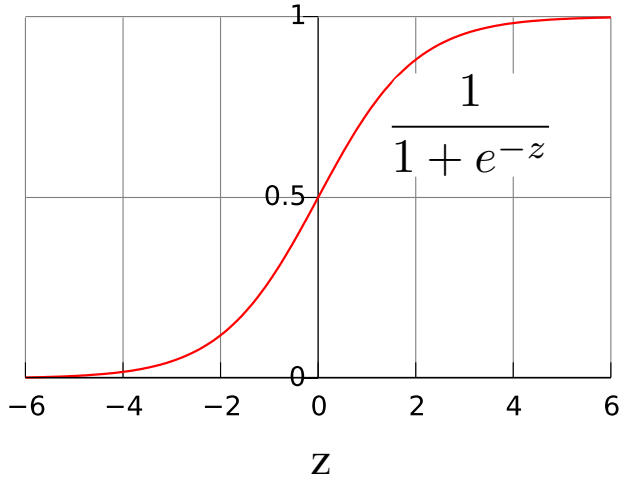
Logistic Regression

- Learn P(Y|**X**) directly!
  - Assume a particular functional form
  - Linear classifier? On one side we say P(Y=1|X)=1, and on the other P(Y=1|X)=0
  - ***But, this is not differentiable (hard to learn)… doesn't allow for label noise...***



Logistic Regression

Logistic function (Sigmoid):



- Learn P(Y|**X**) directly!
  - Assume a particular functional form
  - Sigmoid applied to a linear function of the data:

$$P(Y=1|X) = \frac{1}{1+\exp(w_0+\sum_{i=1}^{n} w_i X_i)}$$

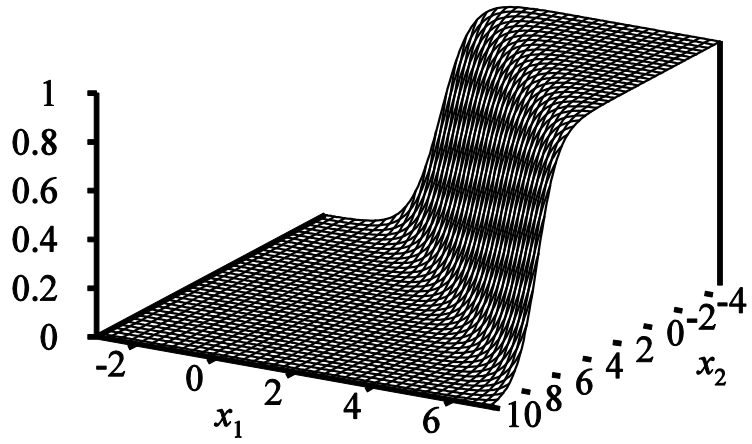
$$P(Y=0|X) = \frac{\exp(w_0+\sum_{i=1}^{n} w_i X_i)}{1+\exp(w_0+\sum_{i=1}^{n} w_i X_i)}$$

Features can be discrete or continuous!

Logistic Function in n Dimensions

$$P(Y = 1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

Sigmoid applied to a linear function of the data:



Features can be discrete or continuous!

Logistic Regression: decision boundary

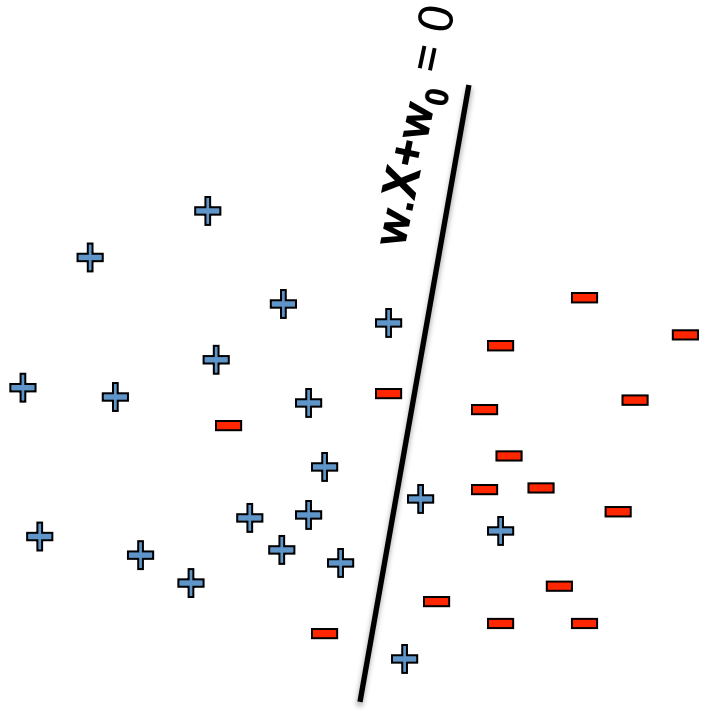
$$P(Y=1|X) = \frac{1}{1+\exp(w_0+\sum_{i=1}^{n} w_i X_i)} \qquad P(Y=0|X) = \frac{\exp(w_0+\sum_{i=1}^{n} w_i X_i)}{1+\exp(w_0+\sum_{i=1}^{n} w_i X_i)}$$

- Prediction: Output the Y with highest P(Y|X)
  - For binary Y, output Y=0 if

$$1 < \frac{P(Y=0|X)}{P(Y=1|X)}$$

$$1 < \exp(w_0 + \sum_{i=1}^{n} w_i X_i)$$

$$0 < w_0 + \sum_{i=1}^{n} w_i X_i$$

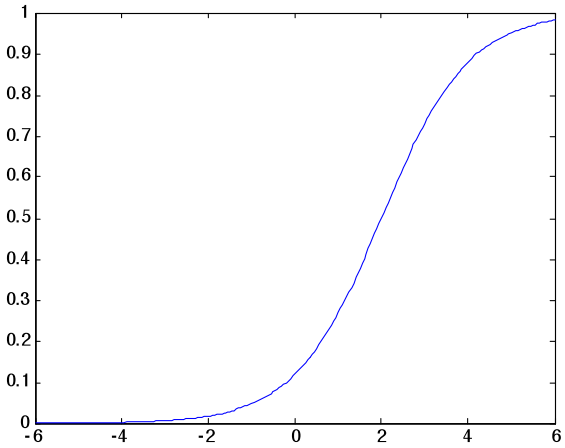


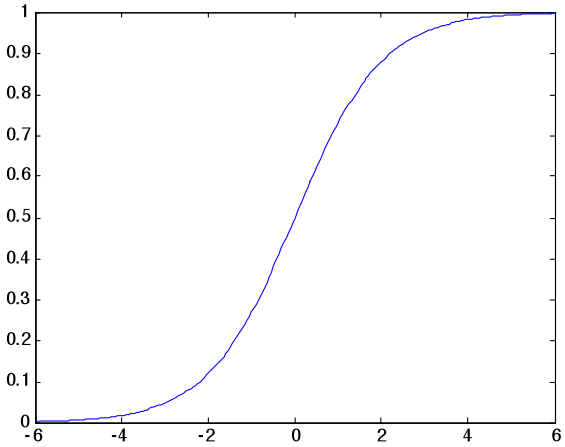
A Linear Classifier!

Understanding Sigmoids

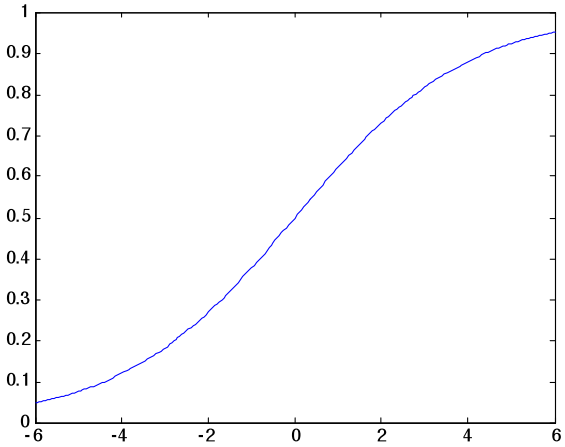
$$g(w_0 + \sum_i w_i x_i) = \frac{1}{1 + e^{w_0 + \sum_i w_i x_i}}$$

w_0=-2, w_1=-1





w_0=0, w_1=-1



w_0=0, w_1= -0.5

Likelihood vs. Conditional Likelihood

Generative (Naïve Bayes) maximizes **Data likelihood**

$$\begin{aligned} \ln P(\mathcal{D} \mid \mathbf{w}) &= \sum_{j=1}^{N} \ln P(\mathbf{x}^j, y^j \mid \mathbf{w}) \\ &= \sum_{j=1}^{N} \ln P(y^j \mid \mathbf{x}^j, \mathbf{w}) + \sum_{j=1}^{N} \ln P(\mathbf{x}^j \mid \mathbf{w}) \end{aligned}$$

Discriminative (Logistic Regr.) maximizes **Conditional Data Likelihood**

$$\ln P(\mathcal{D}_Y \mid \mathcal{D}_{\mathbf{X}}, \mathbf{w}) = \sum_{j=1}^{N} \ln P(y^j \mid \mathbf{x}^j, \mathbf{w})$$

Discriminative models *can't* compute P($\mathbf{x}^j$|**w**)!

Or, … "They don't *waste effort* learning P(**X**)"

Focus only on P(Y|**X**) - all that matters for classification