CS 760: Machine Learning
Naïve Bayes

Kirthi Kandasamy

University of Wisconsin-Madison

February 20, 2023

Announcements

- HW3 due next Monday (Feb 27)

- **Midterm**
  - Will be in-person
  - Thursday 9th March at 7.30pm
  - Room B130 Van Vleck Hall

- **Some outstanding issues from class**
  - Precision-recall curve
  - Objective for logistic regression

Outline

- **Generative and Discriminative Models**
  - Comparison, MAP vs MLE
- **Naïve Bayes**
  - Motivation, Training, Inference, Smoothing
- **Naïve Bayes Examples**
  - Bernoulli, Multiclass, Gaussian

Outline

- **Generative and Discriminative Models**
  - Comparison, MAP vs MLE
- **Naïve Bayes**
  - Motivation, Training, Inference, Smoothing
- **Naïve Bayes Examples**
  - Bernoulli, Multiclass, Gaussian

Supervised Learning: Review

Problem setting

- Set of possible instances $\mathcal{X}$
- Unknown *target function* $f : \mathcal{X} \to \mathcal{Y}$
- Set of *models* (a.k.a. *hypotheses*) $\mathcal{H} = \{h | h : \mathcal{X} \to \mathcal{Y}\}$

Get

- Training set of instances for unknown target function *f*,

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})$$

Goal: model *h* that best approximates *f*

Discriminative Models

- **Idea**: hypothesis h directly predicts the label (given features)
  - y = h(x) or p(y|x) = h(x)

- We saw this already in linear regression & logistic regression
  - Linear regression:

$$h_\theta(x) = \sum_{i=0}^{d} \theta_i x_i$$

  - Logistic regression:

$$P_\theta(y = 1|x) = \sigma(\theta^T x) = \frac{1}{1 + \exp(-\theta^T x)}$$

Generative Models

- (Typically) probabilistic model which states how the data was **generated.**
  $h(x,y) = p(x,y)$ or $h(x) = p(x)$ ← **Note: supervised or unsupervised**

- Select a hypothesis via MLE, MAP etc.
  - Use Bayes' rule to determine predictions

- In discriminative probabilistic models, we model how the labels y were generated conditioned on the features x.
  We do not usually model p(x).

Probabilistic models: Generative vs Discriminative

Generative models	Discriminative models
Can be used for both supervised and unsupervised learning	Typically used only for supervised learning.
Specifies a probabilistic model for how data was generated. p(X) for supervised learning, p(X, Y) for unsupervised learning.	Specifies a probabilistic model for how the labels were generated conditioned on the features for supervised learning, i.e p(Y|X).
In supervised learning, we may model *(i)* p(X, Y) jointly, *(ii)* p(Y) first and then p(X|Y), or *(iii)* p(X) first and then p(Y|X). In my experience *(ii)* is most common.	In supervised learning, we always model p(Y|X)
Use MLE, MAP etc to estimate model parameters.	

Review: **Maximum Likelihood Estimation (MLE)**

- For some set of data, find the parameters that maximize the likelihood / log-likelihood

$$\hat{\theta} = \arg\max_{\theta} \mathcal{L}(\theta; X)$$

- Example: suppose we have n samples from a Bernoulli distribution

$$P_\theta(X = x) = \begin{cases} \theta & x = 1 \\ 1 - \theta & x = 0 \end{cases}$$

Then, if k of the n samples are 1

$$\mathcal{L}(\theta; X) = \prod_{i=1}^{n} P(X = x_i) = \theta^k (1 - \theta)^{n-k} \quad \Longrightarrow \quad \widehat{\theta}_{\mathrm{MLE}} = \frac{k}{n}$$

MLE Example (do at home)

- For some set of data, find the parameters that maximize the likelihood / log-likelihood
- Example: exponential distribution

  - pdf of Exponential(λ): $f(x) = \lambda e^{-\lambda x}$
  - Suppose $X_i \sim$ Exponential(λ) for $1 \leq i \leq N$.
  - Find MLE for data $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$
  - First write down log-likelihood of sample.
  - Compute first derivative, set to zero, solve for λ.
  - Compute second derivative and check that it is concave down at λ^{MLE}.

MLE Example (do at home)

- Example: exponential distribution
  - First write down log-likelihood of sample.

$$\ell(\lambda) = \sum_{i=1}^{N} \log f(x^{(i)}) \tag{1}$$

$$= \sum_{i=1}^{N} \log(\lambda \exp(-\lambda x^{(i)})) \tag{2}$$

$$= \sum_{i=1}^{N} \log(\lambda) + -\lambda x^{(i)} \tag{3}$$

$$= N \log(\lambda) - \lambda \sum_{i=1}^{N} x^{(i)} \tag{4}$$

MLE Example (do at home)

- Example: exponential distribution

  - Compute first derivative, set to zero, solve for λ.

$$\frac{d\ell(\lambda)}{d\lambda} = \frac{d}{d\lambda} N \log(\lambda) - \lambda \sum_{i=1}^{N} x^{(i)} \tag{1}$$

$$= \frac{N}{\lambda} - \sum_{i=1}^{N} x^{(i)} = 0 \tag{2}$$

$$\boxed{\Rightarrow \lambda^{\text{MLE}} = \frac{N}{\sum_{i=1}^{N} x^{(i)}}} \tag{3}$$

Another Approach: **Bayesian Inference**

- Let us consider a different approach
- Need a little bit of terminology

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

- *H* is the hypothesis
- *E* is the evidence

Bayesian Inference Definitions

- Terminology:

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)} \longleftarrow \textbf{Prior}$$

- Prior: estimate of the probability **without** evidence

Bayesian Inference Definitions

- Terminology:

$$P(H|E) = \frac{\overset{\text{Likelihood}}{P(E|H)}P(H)}{P(E)}$$

- Likelihood: probability of evidence **given a hypothesis**.

Bayesian Inference Definitions

- Terminology:

$$\underset{\substack{\uparrow \\ \textbf{Posterior}}}{P(H|E)} = \frac{P(E|H)P(H)}{P(E)}$$

- Posterior: probability of hypothesis **given evidence**.

Maximum a Posteriori (MAP) Estimation

- We treat the parameters of a model as random variables with a *prior* probability distribution.

- Then, treat learning as Bayesian inference
  - "Evidence" is the data

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

  - **Maximum a posteriori probability** (**MAP**) estimation

$$\theta^{\mathrm{MAP}} = \arg\max_{\theta} \prod_{i=1}^{n} p(x^{(i)}|\theta)p(\theta)$$

MAP vs **ML**

- What is the difference between ML and MAP?

$$\theta^{\mathrm{MLE}} = \arg\max_{\theta} \prod_{i=1}^{n} p(x^{(i)}|\theta)$$

$$\theta^{\mathrm{MAP}} = \arg\max_{\theta} \prod_{i=1}^{n} p(x^{(i)}|\theta)p(\theta)$$

  - Prior!

Break & Quiz

Q1-1: Are these statements true or false?
(A) Generative methods model joint probability distribution while discriminative methods model posterior probabilities of Y given X.
(B) We usually train a discriminative model by maximizing the posteriors for true labels for supervised tasks.

1. True, True
2. True, False
3. False, True
4. False, False

Q: Are these statements true or false?
(A) Generative methods model joint probability distribution while discriminative methods model conditional probabilities of Y given X.
(B) We usually train a discriminative model by maximizing the posteriors for true labels for supervised tasks.

1. True, True
2. [✓ True, False]
3. False, True
4. False, False

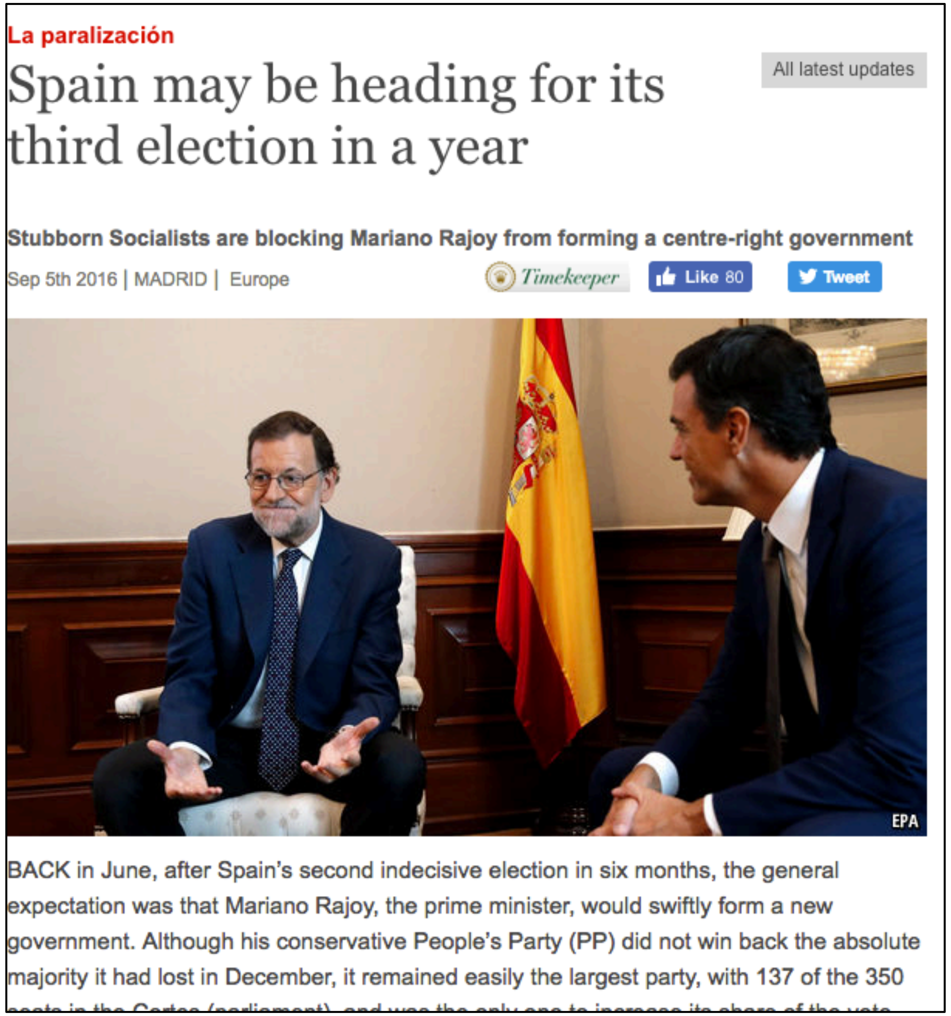
(A) The aim of a generative model is to learn the generative story, i.e. the joint distribution . On the other hand, a discriminative model aims to directly learn the posterior probability .
(B) We usually train a discriminative model by minimizing the corresponding loss function. MLE is also ok, but it often requires us to specify the distribution first, which makes the learning problem more complicated, thus limiting its application area.

Outline

- **Generative and Discriminative Models**
  - Comparison, MAP vs MLE
- **Naïve Bayes**
  - Motivation, Training, Inference, Smoothing
- **Naïve Bayes Examples**
  - Bernoulli, Multiclass, Gaussian

Application: Parody Detection

- The Economist
- The Onion



La paralización

All latest updates

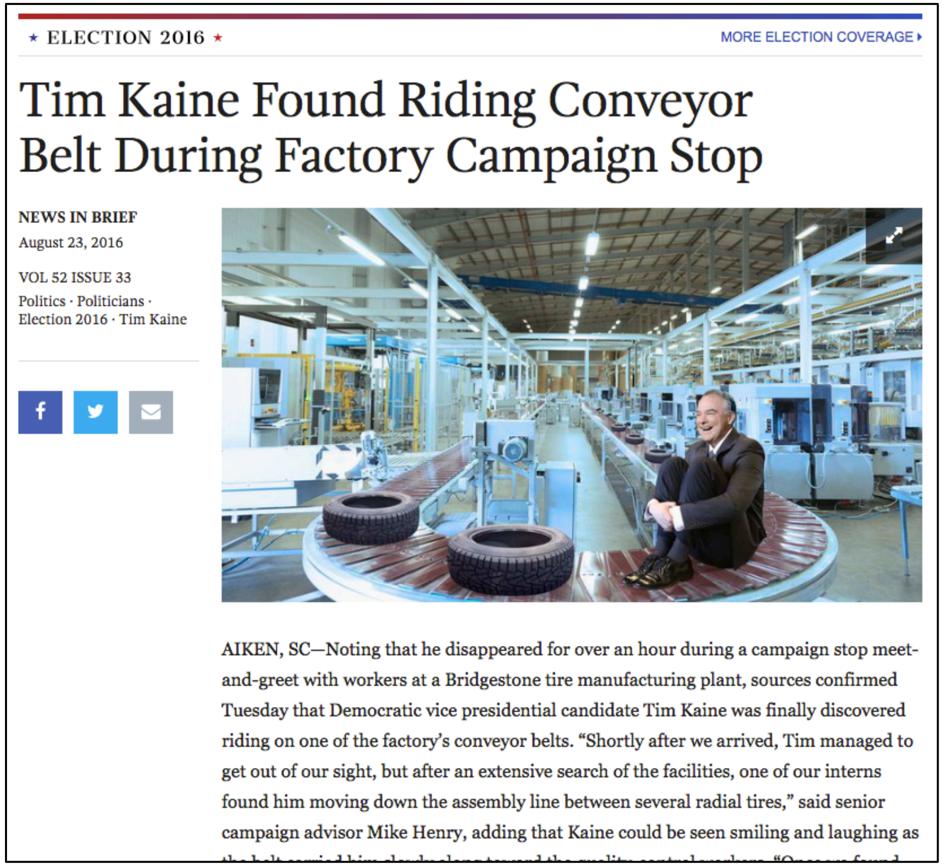
Spain may be heading for its third election in a year

Stubborn Socialists are blocking Mariano Rajoy from forming a centre-right government

Sep 5th 2016 | MADRID | Europe

EPA

BACK in June, after Spain's second indecisive election in six months, the general expectation was that Mariano Rajoy, the prime minister, would swiftly form a new government. Although his conservative People's Party (PP) did not win back the absolute majority it had lost in December, it remained easily the largest party, with 137 of the 350 seats in the Cortes (parliament), and was the only one to increase its share of the vote



★ ELECTION 2016 ★

MORE ELECTION COVERAGE ▸

Tim Kaine Found Riding Conveyor Belt During Factory Campaign Stop

NEWS IN BRIEF
August 23, 2016

VOL 52 ISSUE 33

Politics · Politicians · Election 2016 · Tim Kaine

AIKEN, SC—Noting that he disappeared for over an hour during a campaign stop meet-and-greet with workers at a Bridgestone tire manufacturing plant, sources confirmed Tuesday that Democratic vice presidential candidate Tim Kaine was finally discovered riding on one of the factory's conveyor belts. "Shortly after we arrived, Tim managed to get out of our sight, but after an extensive search of the facilities, one of our interns found him moving down the assembly line between several radial tires," said senior campaign advisor Mike Henry, adding that Kaine could be seen smiling and laughing as the belt carried him slowly along toward the quality control workers. "Once we found

Model 0: Not-Naïve Model

Generative model:

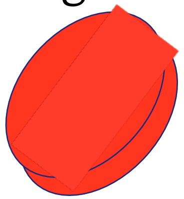
1. Flip a weighted coin (Y)
2. If heads, sample a document (X) from the Spam distribution
3. If tails, sample a document (X) from the Not-Spam distribution

$$P(X, Y) = P(X|Y)P(Y)$$

4. We first sample Y, then sample X given Y.

Model 0: Not-Naïve Model using Bag of words representation

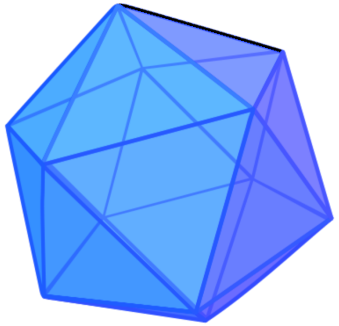
Flip weighted coin



If HEADS, roll gray die

y	x_1	x_2	x_3	…	x_K
0	1	0	1	…	1
1	0	1	0	…	1
1	1	1	1	…	1
0	0	0	1	…	1
0	1	0	1	…	0
1	1	0	1	…	0

If TAILS, roll blue die



Each side of the die is labeled with a document vector (e.g. [1,0,1,…,1])

Model 0: Not-Naïve Model using Bag of words representation

Generative model:

1. Flip a weighted coin (Y)
2. If heads, roll the **gray** many sided die to sample a bag of words document vector ($\boldsymbol{X}$) from the Spam distribution
3. If tails, roll the **blue** many sided die to sample a bag of words document vector ($\boldsymbol{X}$) from the Not-Spam distribution

$$P(X_1, \ldots, X_K, Y) = P(X_1, \ldots, X_K | Y) P(Y)$$

Model 0: Main Problem

How many terms are we modeling?

- Say K is the number of words in a dictionary and the features are binary (a word exists or not): $X_i \in \{0, 1\}$

$$P(X_1, \ldots, X_K | Y)$$

- 2^k choices of feature vector, each gets its own probability…
  - Exponentially big table (in feature vector size)

Naïve Bayes: Core Assumption

How do we fix this problem?

- Conditional **independence** of features:

$$P(X_1, \ldots, X_K, Y) = P(X_1, \ldots, X_K | Y) P(Y)$$

$$= \left(\prod_{k=1}^{K} P(X_k | Y) \right) P(Y)$$

  - What do we gain? With binary features, get 2 entries per feature
  - So, number of probabilities $2^k \rightarrow 2k$

Break & Quiz

Q2-1: Are these statements true or false?
(A) Naïve Bayes assumes conditional independence of features to decompose the joint probability into the conditional probabilities.
(B) We can use Naïve Bayes' to reduce model complexity which helps with over-fitting

1. True, True
2. True, False
3. False, True
4. False, False

Q2-1: Are these statements true or false?
(A) Naïve Bayes assumes conditional independence of features to decompose the joint probability into the conditional probabilities.
(B) We can use Naïve Bayes' to reduce model complexity which helps with over-fitting

1. [✓ True, True]
2. True, False
3. False, True
4. False, False

(A) Just as we learnt in the lecture.
(B) True, since the fully-fledged joint model subsumes the conditionally independent model.

Outline

- **Generative and Discriminative Models**
  - Comparison, MAP vs MLE
- **Naïve Bayes**
  - Motivation, Training, Inference, Smoothing
- **Naïve Bayes Examples**
  - Bernoulli, Multiclass, Gaussian

Naïve Bayes: Overall Model

Support: Depends on the problem, $P(X_k|Y)$

Model: Product of **prior** and the event model

$$P(\mathbf{X}, Y) = P(Y) \prod_{k=1}^{K} P(X_k|Y)$$

Training: Find the **class-conditional** MLE parameters

For $P(Y)$, we find the MLE using the data. For each $P(X_k|Y)$ we condition on the data with the corresponding class.

Prediction: Find the class that maximizes the posterior

$$\hat{y} = \operatorname*{argmax}_{y} p(y|\mathbf{x})$$

Naïve Bayes: Predicting

- With conditional probabilities, how to predict?

$$\hat{y} = \underset{y}{\operatorname{argmax}}\, p(y|\mathbf{x}) \quad \text{(posterior)}$$

$$= \underset{y}{\operatorname{argmax}}\, \frac{p(\mathbf{x}|y)p(y)}{p(x)} \quad \text{(by Bayes' rule)}$$

$$= \underset{y}{\operatorname{argmax}}\, p(\mathbf{x}|y)p(y)$$

Naïve Bayes Example 1: Bernoulli

Support: Binary vectors of length K

$$\mathbf{x} \in \{0,1\}^K$$

Generative Model:

$$Y \sim \text{Bernoulli}(\phi)$$

$$X_k \sim \text{Bernoulli}(\theta_{k,Y}) \; \forall k \in \{1, \ldots, K\}$$

Joint probability :

$$\begin{aligned} p_{\phi,\boldsymbol{\theta}}(\boldsymbol{x}, y) &= p_{\phi,\boldsymbol{\theta}}(x_1, \ldots, x_K, y) \\ &= p_\phi(y) \prod_{k=1}^{K} p_{\boldsymbol{\theta}_k}(x_k|y) \\ &= (\phi)^y (1-\phi)^{(1-y)} \prod_{k=1}^{K} (\theta_{k,y})^{x_k} (1-\theta_{k,y})^{(1-x_k)} \end{aligned}$$

Training Bernoulli Naïve Bayes

- Recall: train (by MLE) is to find **class-conditional** parameters

- To find P(Y): use all the data
  - For P(X_i|Y=y): use the data for that class

$$\phi = \frac{\sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1)}{N}$$

$$\theta_{k,0} = \frac{\sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0 \wedge x_k^{(i)} = 1)}{\sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0)}$$

$$\theta_{k,1} = \frac{\sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1 \wedge x_k^{(i)} = 1)}{\sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1)}$$

$$\forall k \in \{1, \ldots, K\}$$

Naïve Bayes: Training

- **Training**: empirically estimate the probabilities
  - Store: conditional probability tables (CPTs)

Y	P(Y)
0	0.33
1	0.67

X_1	Y	P(X_1|Y)
0	0	0.1
0	1	0.9
1	0	0.9
1	1	0.1

X_2	Y	P(X_2|C)
0	0	0.2
0	1	0.5
1	0	0.8
1	1	0.5

Naïve Bayes: Smoothing

- **Training**: empirically estimate the probabilities
  - We are just obtaining counts to estimate $P(X_i|Y)$
  - Suppose X_i has K possible values, and our counts for Y=y are $b_1,\ldots,b_K$
  - **What if $b_i = 0$?**
    - Predictions will end up being zero. We want to prevent this (why?).
  - Solution: smooth!

$$\widehat{P}(X_i = j|Y = y) = \frac{b_j + \alpha}{\sum_k b_k + \alpha K}$$

Naïve Bayes Example 2: Multinomial

Support: multinomial vectors of length d

Generative model: (for each data point)

- Generate label: $y \sim \text{Mult}(\phi), \quad \text{where} \quad \sum_{k=1}^{K} \phi_k = 1$

- For each feature,

$$x_i \sim \text{Mult}(\theta_{i,y}), \quad \text{where} \quad \sum_{k=1}^{K_i} \theta_{i,y,k} = 1$$

Joint probability:

$$p_{\phi,\theta}(x, y) = p_\phi(y) \prod_{i=1}^{d} p_{\theta_{i,y}}(x_i | y) = \phi_y \prod_{i=1}^{d} \theta_{i,y,x_i}$$

Naïve Bayes Example 3: Gaussian

Support: $\mathbf{x} \in \mathbb{R}^K$

Model: Product of **prior** and the event model

$$p(\boldsymbol{x}, y) = p(x_1, \ldots, x_K, y)$$
$$= p(y) \prod_{k=1}^{K} p(x_k | y)$$

Gaussian Naive Bayes assumes that $p(x_k|y)$ is given by a Normal distribution.

Thanks Everyone!

Some of the slides in these lectures have been adapted/borrowed from materials developed by Mark Craven, David Page, Jude Shavlik, Tom Mitchell, Nina Balcan, Elad Hazan, Tom Dietterich, Pedro Domingos, Jerry Zhu, Yingyu Liang, Volodymyr Kuleshov, and Fred Sala