Naïve Bayes

Instructor: Junghye Lee

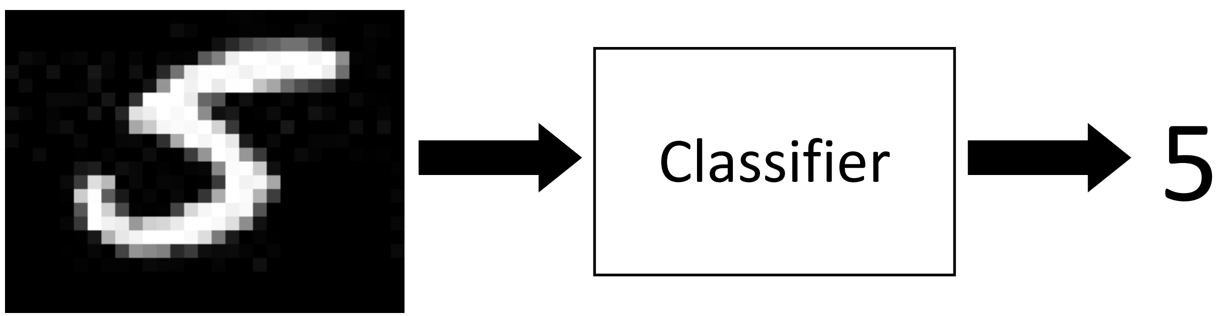
Department of Industrial Engineering
junghyelee@unist.ac.kr

Things We'd Like to Do

- Spam Classification
  - Given an email, predict whether it is spam or not
- Medical Diagnosis
  - Given a list of symptoms, predict whether a patient has cancer or not
- Weather
  - Based on temperature, humidity, etc… predict if it will rain tomorrow

Bayesian Classification

- Problem statement:
  - Given features $X_1, X_2, \dots, X_n$
  - Predict a label Y
  - E.g.) Digit recognition



  - $X_1, X_2, \dots, X_n \in \{0,1\}$ (Black vs. White pixels)
  - $Y \in \{5,6\}$ (predict whether a digit is a 5 or a 6)

Bayes Theorem

$$\overset{\text{Likelihood}}{\downarrow}\quad\overset{\text{Prior}}{\downarrow}$$

$$p(A|B) = \frac{\overbrace{p(B|A)}^{\text{Likelihood}}\,\overbrace{P(A)}^{\text{Prior}}}{\underbrace{p(B)}_{\text{Normalization Constant}}}$$

- $P(A|B) \leftarrow P(Y|\boldsymbol{X})$
- $P(B|A) \leftarrow P(\boldsymbol{X}|Y)$
- $P(A) \leftarrow P(Y)$
- $P(B) \leftarrow P(\boldsymbol{X}) = \Sigma_Y P(X|Y)P(Y)$

Generative model
vs.
Discriminative model

The Bayes Classifier

- In class, we saw that a good strategy is to predict:

$$\text{argmax}_Y P(Y|X_1, \ldots, X_n)$$ **Maximum A Posterior (MAP)**

  - (for example: what is the probability that the image represents a 5 given its pixels?)
  - Posterior probability

- How do we compute that?

The Bayes Classifier

- Use Bayes Rule!

Likelihood | Prior

$$P(Y|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y)P(Y)}{P(X_1,\ldots,X_n)}$$

Normalization Constant

- Why did this help? Well, we think that we might be able to specify how features are “generated” by the class label

The Bayes Classifier

- Let's expand this for our digit recognition task:

$$\begin{aligned}
P(Y=5|X_1,\dots,X_n) &= \frac{P(X_1,\dots,X_n|Y=5)P(Y=5)}{P(X_1,\dots,X_n|Y=5)P(Y=5)+P(X_1,\dots,X_n|Y=6)P(Y=6)} \\
P(Y=6|X_1,\dots,X_n) &= \frac{P(X_1,\dots,X_n|Y=6)P(Y=6)}{P(X_1,\dots,X_n|Y=5)P(Y=5)+P(X_1,\dots,X_n|Y=6)P(Y=6)}
\end{aligned}$$

- To classify, we'll simply compute these two probabilities and predict based on which one is greater

Model Parameters

- For the Bayes classifier, we need to “learn” two functions, the likelihood and the prior

Likelihood: $P(X_1, \dots, X_n | Y = 5), P(X_1, \dots, X_n | Y = 6)$

Prior: $P(Y = 5), P(Y = 6)$

- How many parameters are required to specify the prior for our digit recognition example?

$\mathbf{1}$

- How many parameters are required to specify the likelihood?
  - (Supposing that each image is 30x30 pixels)

$\mathbf{2 \cdot (2^{30\times 30} - 1)}$

Model Parameters

- The problem with explicitly modeling $P(X_1, \ldots, X_n|Y)$ is that there are usually way too many parameters:
  - We'll run out of space
  - We'll run out of time
  - We'll need tons of training data (which is usually not available)

Naïve Bayes Assumption

- Assume that each feature is independent from one another given the class label
- Definition: x is conditionally independent of y given z, if the probability distribution governing x is independent of the value of y, given the value of z
- For example:

  $p(thunder|raining, lightening) = p(thunder|lightening)$
- If x, y are conditional independent given z, we have:

$$p(x, y|z) = p(x|z)p(y|z)$$

The Naïve Bayes Model

- The *Naïve Bayes Assumption*: Assume that **all features are independent given the class label Y: Conditional independence**
- Equationally speaking:

$$P(X_1, \ldots, X_n|Y) = \prod_{i=1}^{n} P(X_i|Y)$$

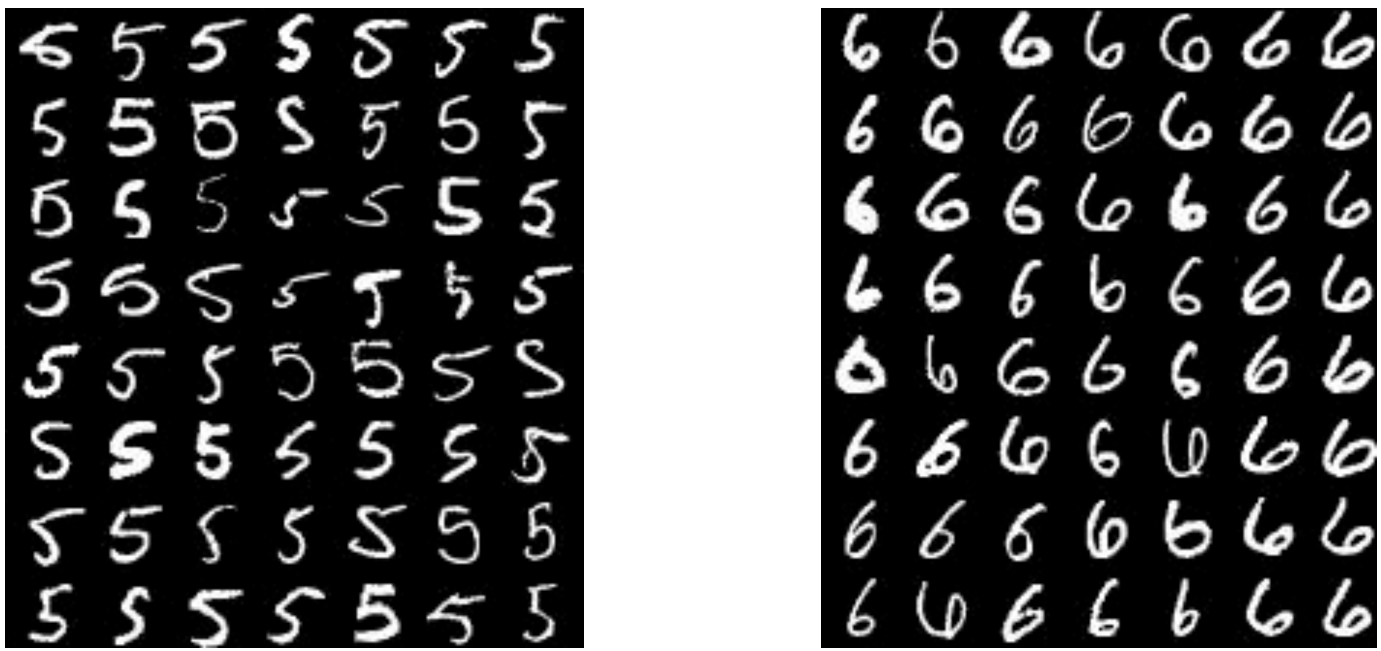
- (We will discuss the validity of this assumption later)

Why is this useful?

- # of parameters for modeling $P(X_1, \dots, X_n | Y)$:
  - $2 * (2^n - 1)$
- # of parameters for modeling $P(X_1|Y), P(X_2|Y), \dots, P(X_n|Y)$
  - $2 * n$

Naïve Bayes Training

- Now that we've decided to use a Naïve Bayes classifier, we need to train it with some data:



MNIST Training Data

Naïve Bayes Training

- Training in Naïve Bayes is **easy**:
  - Estimate $P(Y = v)$ as the fraction of records with $Y = v$

$$P(Y = v) = \frac{Count(Y = v)}{\#\ records}$$

  - Estimate $P(X_i = u|Y = v)$ as the fraction of records with $Y = v$ for which $X_i = u$

$$P(X_i = u|Y = v) = \frac{Count(X_i = u \wedge Y = v)}{Count(Y = v)}$$

- This corresponds to Maximum Likelihood estimation of model parameters.

What if continuous $\boldsymbol{X}$

- K different classes
  - Conditional normal distribution

$$K * (2n)$$

$$P(X_1|Y=1) = \frac{1}{\sqrt{2\pi}\sigma_{11}} \exp\left(-\frac{(x-\mu_{11})^2}{\sigma_{11}}\right)$$

  - How do you calculate μ_{11} and σ_{11}?

Naïve Bayes Training

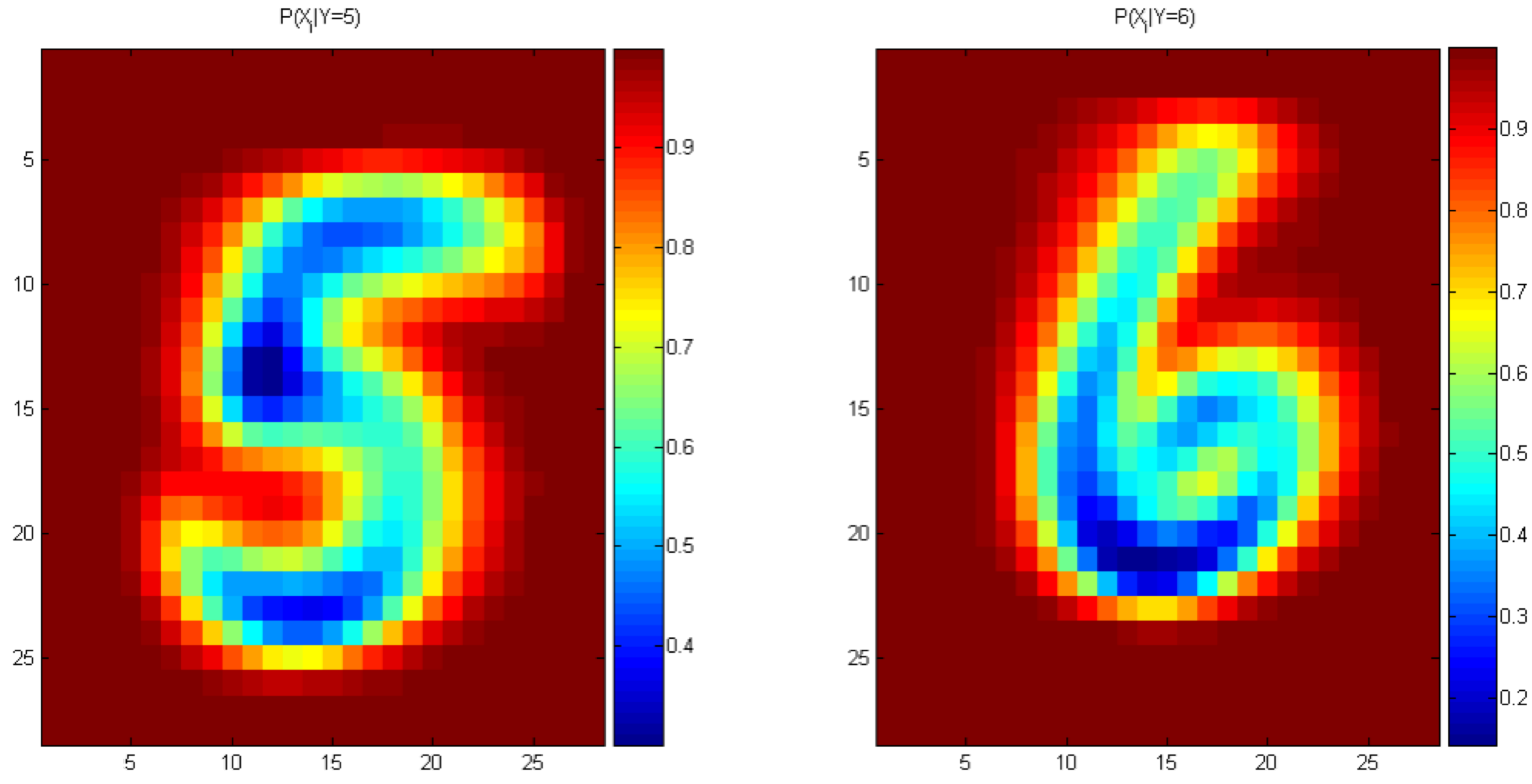
- In practice, some of these counts can be zero
- Fix this by adding "virtual" counts:

$$P(X_i = u|Y = v) = \frac{Count(X_i = u \wedge Y = v) + 1}{Count(Y = v) + 2}$$

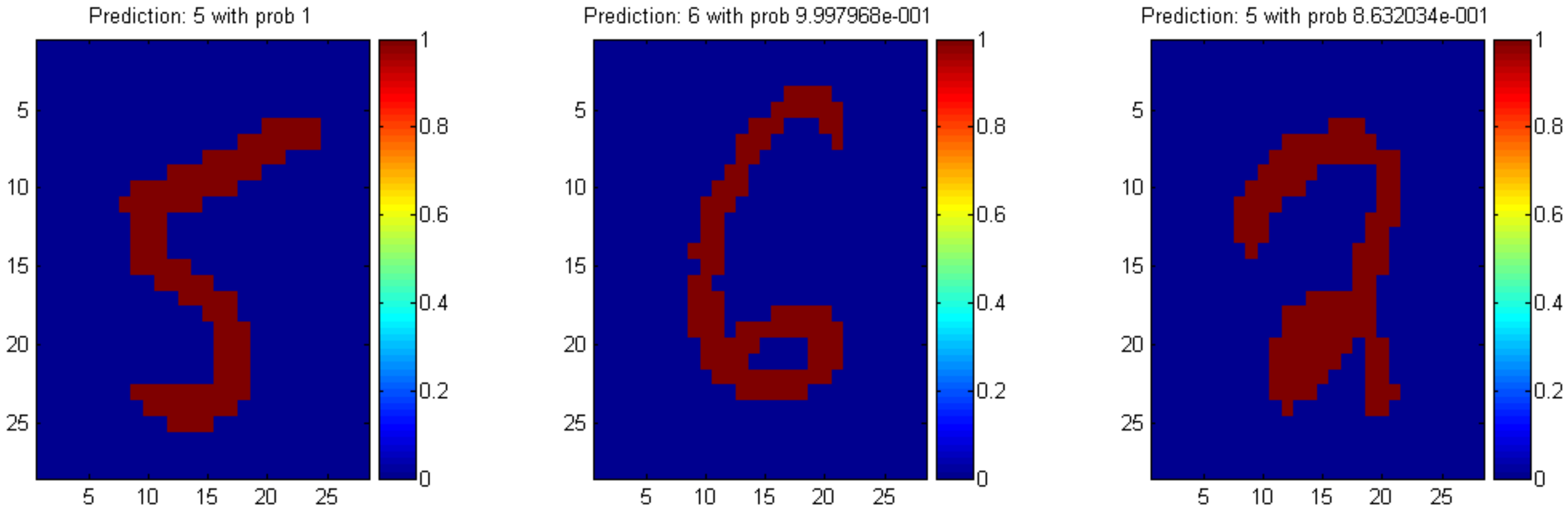
  - (This is like putting a prior on parameters and doing MAP estimation instead of MLE)
  - This is called *Smoothing*

Naïve Bayes Training

- For binary digits, training amounts to averaging all of the training fives together and all of the training sixes together.



Naïve Bayes Classification



Naïve Bayes Example

The weather data, with counts and probabilities													
outlook			temperature			humidity			windy			play	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

A new day				
outlook	temperature	humidity	windy	play
sunny	cool	high	true	?

Naïve Bayes Example

- Likelihood of yes $=\frac{2}{9}\times\frac{3}{9}\times\frac{3}{9}\times\frac{3}{9}\times\frac{9}{14}=0.0053$
- Likelihood of no $=\frac{3}{5}\times\frac{1}{5}\times\frac{4}{5}\times\frac{3}{5}\times\frac{5}{14}=0.0206$
- That’s it?
- $Likelihood * Prior$
- $P(Y = yes) = \frac{9}{14}, P(Y = no) = \frac{5}{14}$
- Therefore, the prediction is No

The numeric weather data with summary statistics													
outlook			temperature			humidity			windy			play	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3		83	85		86	85	false	6	2	9	5
overcast	4	0		70	80		96	90	true	3	3		
rainy	3	2		68	65		80	70					
				64	72		65	95					
				69	71		70	91					
				75			80						
				75			70						
				72			90						
				81			75						
sunny	2/9	3/5	mean	73	74.6	mean	79.1	86.2	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	std dev	6.2	7.9	std dev	10.2	9.7	true	3/9	3/5		
rainy	3/9	2/5											

Naïve Bayes Example

- For examples,

$$f(\text{temperature}=66 \mid \text{Yes}) = \frac{1}{\sqrt{2\pi}(6.2)} e^{-\frac{(66-73)^2}{2(6.2)^2}} = 0.0340$$

- Likelihood of Yes = $\frac{2}{9} \times 0.0340 \times 0.0221 \times \frac{3}{9} \times \frac{9}{14} = 0.000036$

- Likelihood of No = $\frac{3}{5} \times 0.0291 \times 0.038 \times \frac{3}{5} \times \frac{5}{14} = 0.000136$

Assumption

- Actually, the Naïve Bayes assumption is almost never true
  - E.g.) XOR problem

X_1	X_2	$P(Y=0\|X_1,X_2)$	$P(Y=1\|X_1,X_2)$
0	0	1	0
0	1	0	1
1	0	0	1
1	1	1	0

Counter example?

Nevertheless…

- Naïve Bayes often performs surprisingly well even when its assumptions do not hold.
- Naïve Bayes is often a good choice if you don’t have much training data!
- What’s nice about Naïve Bayes (and generative models in general) is that it returns probabilities.
  - These probabilities can tell us how confident the algorithm is.

Conclusions

- Naïve Bayes is:
  - Really easy to implement and often works well
  - Often a good first thing to try
  - Commonly used as a “punching bag” for smarter algorithms