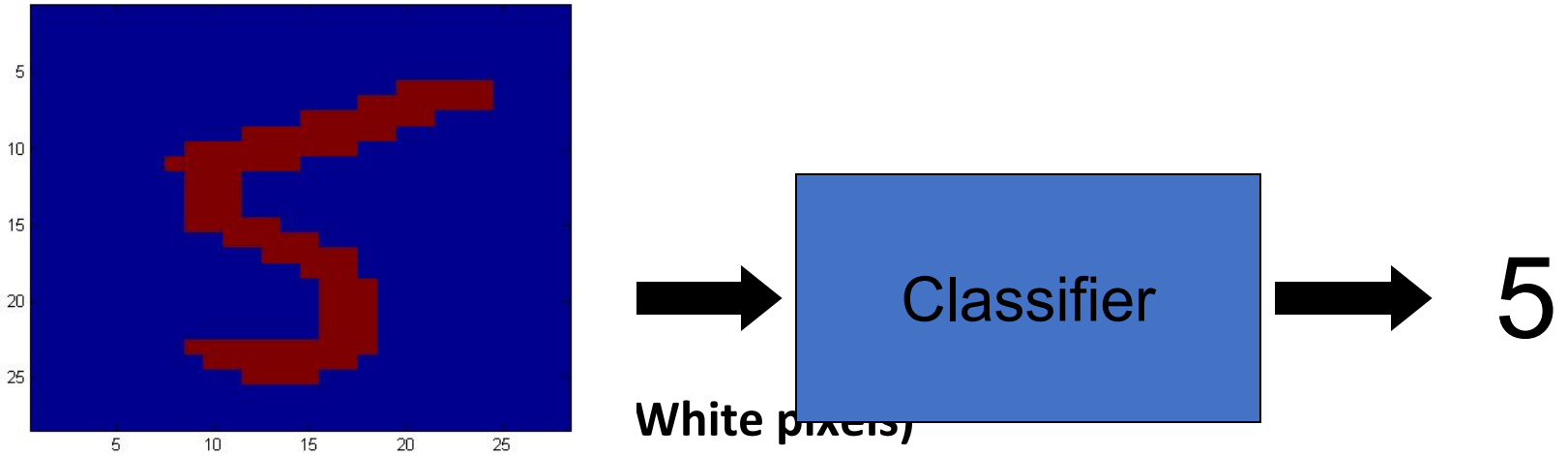
# Another Application

- **Digit Recognition (5 or 6)**




**White pixels)**

- **Y ∈ {5,6} (predict whether a digit is a 5 or a 6)**

# The Bayes Classifier

- A good strategy is to predict:

X: Collection of pixel values

$$\arg\max_Y P(Y|X_1,\ldots,X_n)$$

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

- So … How do we compute that?

# The Bayes Classifier

- Use Bayes Rule!

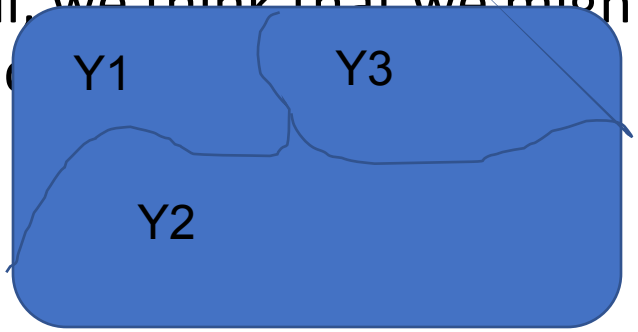
Likelihood

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 = Total probability of feature set

- Why did this help? Well, we think that we might be able to specify how features are "generated" by the c




X space

# The Bayes Classifier

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

$$P(Y = 5|X_1, \ldots, X_n) = \frac{P(X_1, \ldots, X_n|Y = 5)P(Y = 5)}{P(X_1, \ldots, X_n|Y = 5)P(Y = 5) + P(X_1, \ldots, X_n|Y = 6)P(Y = 6)}$$

$$P(Y = 6|X_1, \ldots, X_n) = \frac{P(X_1, \ldots, X_n|Y = 6)P(Y = 6)}{P(X_1, \ldots, X_n|Y = 5)P(Y = 5) + P(X_1, \ldots, X_n|Y = 6)P(Y = 6)}$$

- 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

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

# Model Parameters

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

?

# 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
  - And we'll need tons of training data (which is usually not available)

# The Naïve Bayes Model

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

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

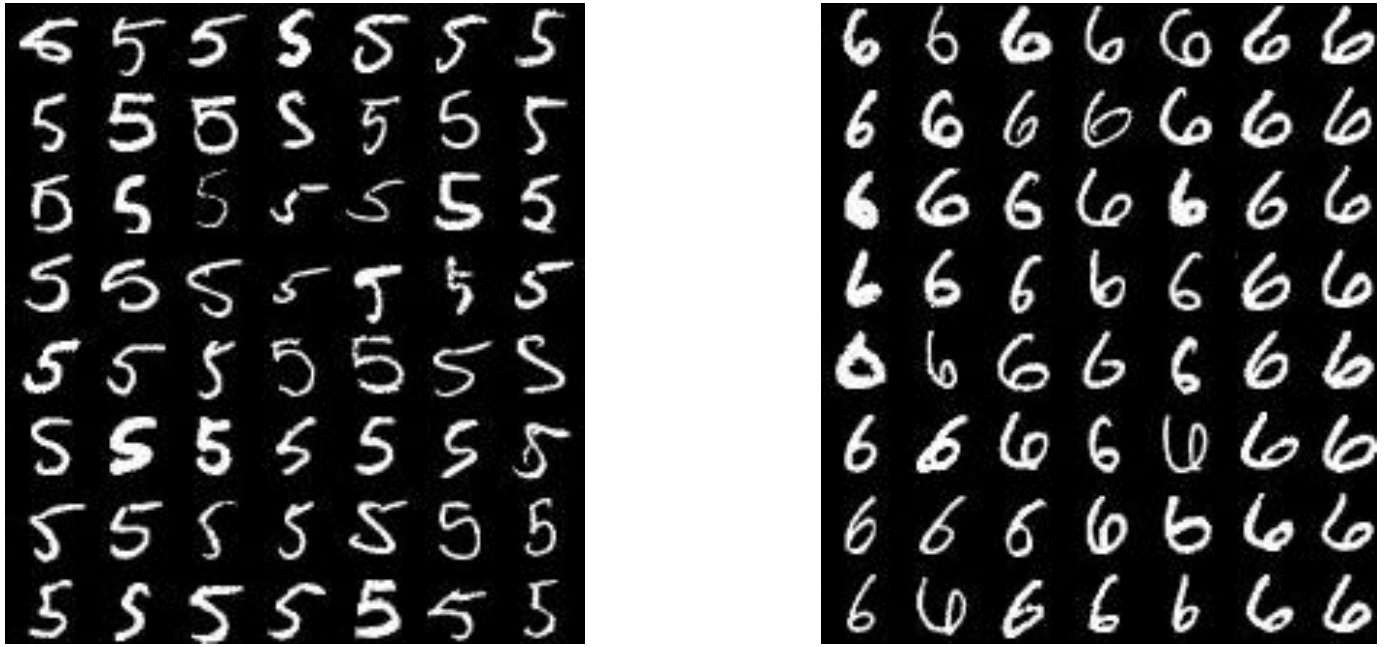
- (We will discuss the validity of this assumption later)

# Why is this useful?

- # of likelihoods for modeling $P_k(X_1,\ldots,X_n|Y)$
- K Classes and n features:
  - $K(2^n) = 2*2^{900}$ Likelihoods
- # of parameters for modeling $P(X_1|Y),\ldots,P(X_n|Y)$
  - Kn (2*900 Likelihoods)
  - K Priors

# Naïve Bayes Training

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



MNIST Training Data

# Naïve Bayes Training

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

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

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

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

# Naïve Bayes Training - smoothing

- In practice, some of these probabilities/ counts can be zero

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

m*p

m

  - m = Number of values the parameter may take
  - p probability of $i^{th}$ parameter value (1/m if equiprobable)

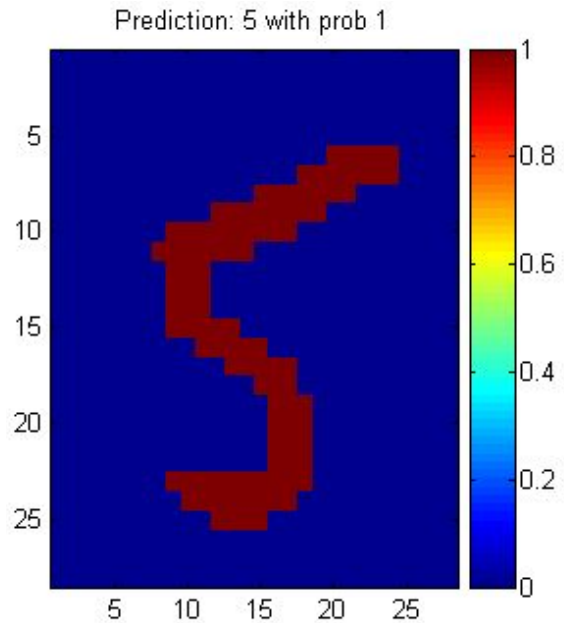
# Smoothing

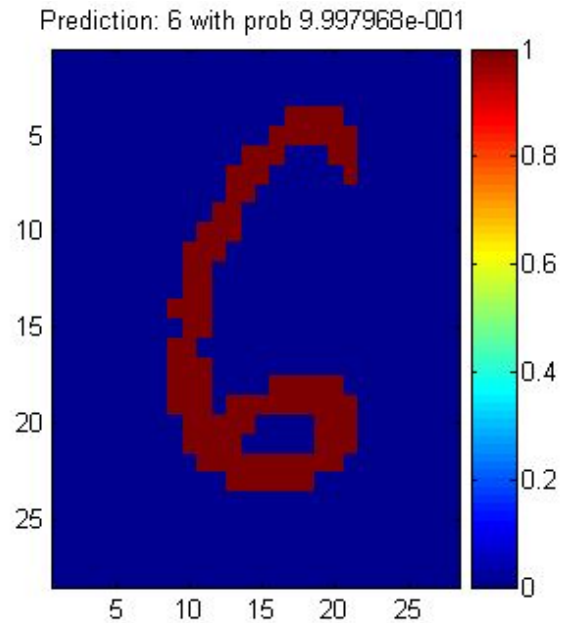
- For Text data, m = |Vocabulary|
- $P = \frac{1}{Vocabulary}$

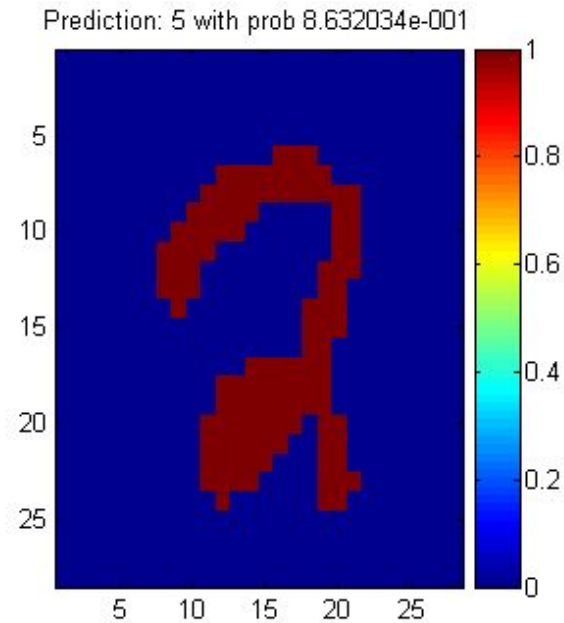
# Color Images NB Training

- For binary digits, how many pixel values are there ?
- training amounts to either
  - finding probabilities of each pixel being  R,G,B for each class
  - finding normal distribution averages and std dev for each of R,G,B values for each class

# Naïve Bayes Classification



# Another Example of the Naïve Bayes Classifier

**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 | ? |

- 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

$$= \frac{3}{5} \times \frac{1}{5} \times \frac{4}{5} \times \frac{3}{5} \times \frac{5}{14} = 0.0206$$

- Therefore, the prediction is No

# The Naive Bayes Classifier for Data Sets with Numerical Attribute Values

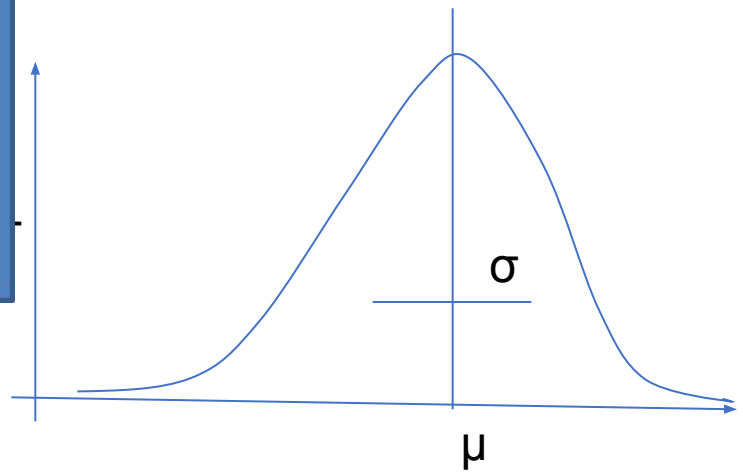
- One common practice to handle numerical attribute values is to assume normal distributions for numerical attributes.

| 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 | | | | | | | | | | | |

# TWO WAYS TO HANDLE CONTINUOUS VALUED ATTRIBUTES

- Disretize
- Assume and calculate mean and standard deviation of Normal distribution from training data

$$\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



# Deriving Normal Distribution

- Let $x_1, x_2, \ldots, x_n$ be the values of a numerical attribute in the training data set.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\sigma = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2$$

$$f(w) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(w-\mu)^2}{\sigma^2}}$$

Given a new case: Outlook = sunny, temperature = 66, humidity = 80 , wind = true. Posterior probabilities?

- 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 = $\frac{3}{5} \times 0.0291 \times 0.038 \times \frac{3}{5} \times \frac{5}{14} = 0.000136$

Total Prob of Yes = 36/(36+136)= 26.47%
Total Prob of No = 136/(36+136)= 73.53%

# Outputting Probabilities

- 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
  - Such a confidence level is not immediately present in DT

# Comparison of DT and NB

## DT

1. Greedy heuristic
2. Discriminative model, cant generate data.
3. Automatic feature prioritization
4. Overfitting - Need pruning / stop growth
5. Support at leaves
6. No issue with disappearance of values
7. No assumption of  independence of features
8. Discretization of continuous values needed
9. Good with lots of data

## NB

1. Statistical
2. Generative model  (calculates prob dist and can generate data) and need Bayes theorem to calculate a-posteriors
3. Manual feature selection
4. No Need for pruning or post training tuning
5. Probabilities show confidence level
6. Can suffer vanishing probs of likelihoods - smoothing
7. NB assumption is there
8. Prob distribution can take care of real values
9. Good with low amounts of data