Introduction to Bayesian methods (continued) - Lecture 16

David Sontag
New York University

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

Outline of lectures

- Review of probability

(After midterm)

Maximum likelihood estimation

2 examples of Bayesian classifiers:

- Naïve Bayes
- Logistic regression

Bayes' Rule

- Two ways to factor a joint distribution over two variables:

$$P(x,y) = P(x|y)P(y) = P(y|x)P(x)$$

- Dividing, we get:

$$P(x|y) = \frac{P(y|x)}{P(y)}P(x)$$

- Why is this at all helpful?
  - Let's us build one conditional from its reverse
  - Often one conditional is tricky but the other one is simple
  - Foundation of many practical systems (e.g. ASR, MT)

- In the running for most important ML equation!

Returning to thumbtack example…

- P(Heads) = θ, P(Tails) = 1-θ

- Flips are *i.i.d.*: $D=\{x_i \mid i=1\ldots n\},\ P(D \mid \theta) = \Pi_i P(x_i \mid \theta)$
  - Independent events
  - Identically distributed according to Bernoulli distribution
- Sequence *D* of α_H Heads and α_T Tails

$$P(\mathcal{D} \mid \theta) = \theta^{\alpha_H}(1-\theta)^{\alpha_T}$$

Called the “likelihood” of the data under the model

Maximum Likelihood Estimation

- **Data:** Observed set D of α_H Heads and α_T Tails
- **Hypothesis:** Bernoulli distribution
- **Learning:** finding θ is an optimization problem
  - What's the objective function?

$$P(\mathcal{D} \mid \theta) = \theta^{\alpha_H}(1-\theta)^{\alpha_T}$$

- **MLE:** Choose θ to maximize probability of *D*

$$\begin{aligned} \widehat{\theta} &= \arg\max_{\theta} \; P(\mathcal{D} \mid \theta) \\ &= \arg\max_{\theta} \; \ln P(\mathcal{D} \mid \theta) \end{aligned}$$

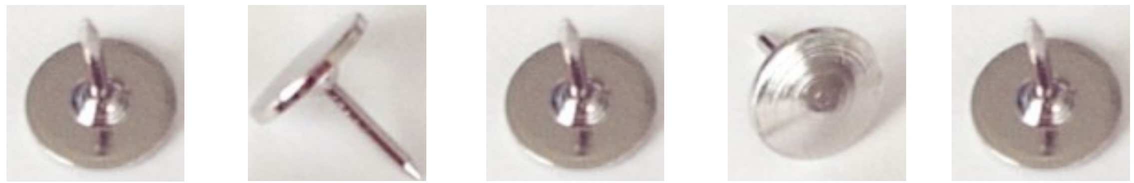
Your first parameter learning algorithm

$$\widehat{\theta} = \arg\max_{\theta} \ln P(\mathcal{D} \mid \theta)$$
$$= \arg\max_{\theta} \ln \theta^{\alpha_H}(1-\theta)^{\alpha_T}$$

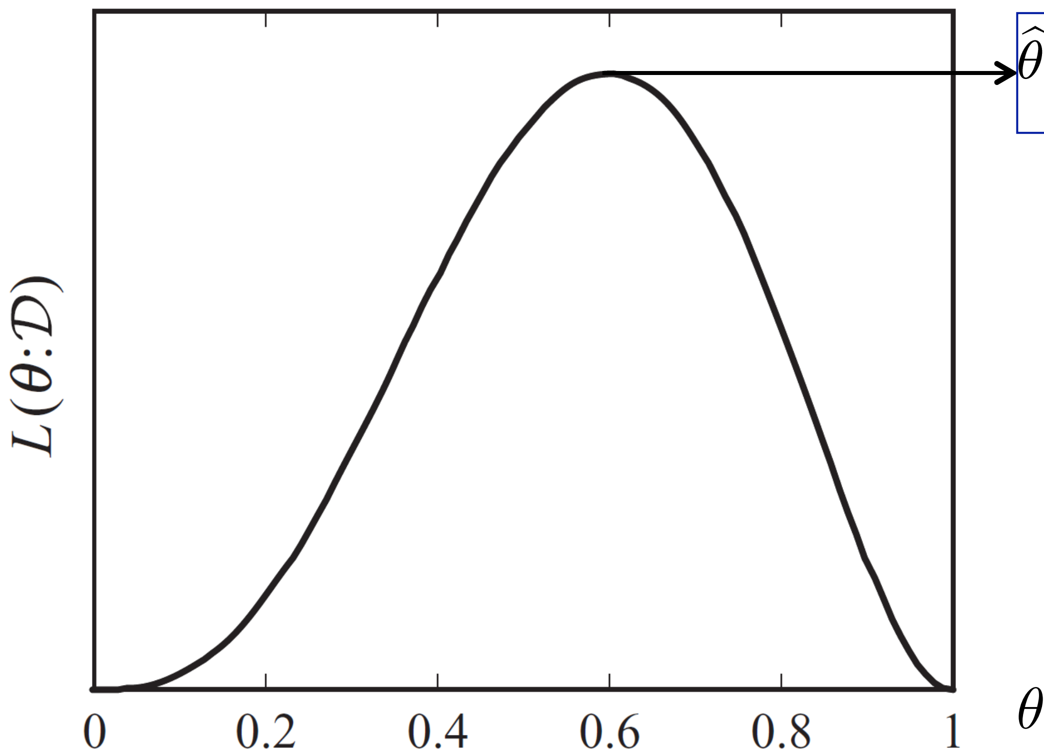
- Set derivative to zero, and solve!

$$\frac{d}{d\theta} \ln P(\mathcal{D} \mid \theta) = \frac{d}{d\theta} \left[\ln \theta^{\alpha_H}(1-\theta)^{\alpha_T}\right]$$
$$= \frac{d}{d\theta} \left[\alpha_H \ln \theta + \alpha_T \ln(1-\theta)\right]$$
$$= \alpha_H \frac{d}{d\theta} \ln \theta + \alpha_T \frac{d}{d\theta} \ln(1-\theta)$$
$$= \frac{\alpha_H}{\theta} - \frac{\alpha_T}{1-\theta} = 0$$

$$\widehat{\theta}_{MLE} = \frac{\alpha_H}{\alpha_H + \alpha_T}$$

Data 

$$L(\theta; \mathcal{D}) = \ln P(\mathcal{D}|\theta)$$

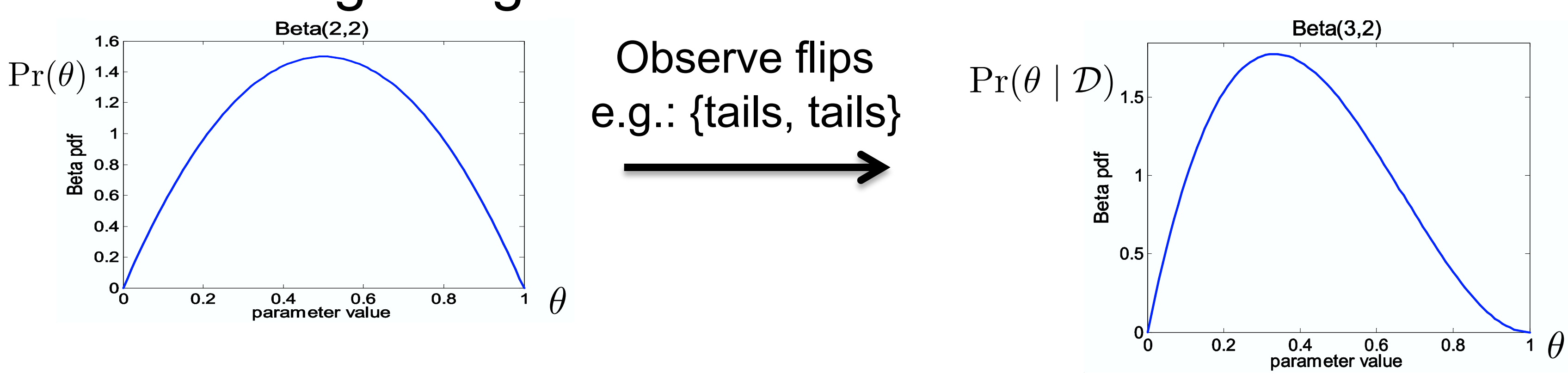


$$\widehat{\theta}_{MLE} = \frac{\alpha_H}{\alpha_H + \alpha_T}$$

What if I have prior beliefs?

- Billionaire says: Wait, I know that the thumbtack is "close" to 50-50. What can you do for me now?
- **You say: I can learn it the Bayesian way…**
- Rather than estimating a single θ, we obtain a distribution over possible values of θ

In the beginning | After observations

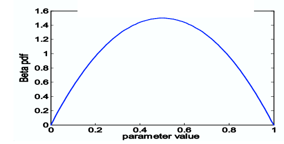


Bayesian Learning

- Use Bayes' rule!

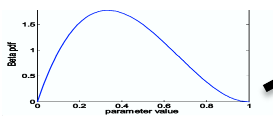
Data Likelihood

Prior



$$P(\theta \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \theta) P(\theta)}{P(\mathcal{D})}$$

Posterior



Normalization

- Or equivalently: $P(\theta \mid \mathcal{D}) \propto P(\mathcal{D} \mid \theta) P(\theta)$
- For *uniform* priors, this reduces to maximum likelihood estimation!

$$P(\theta) \propto 1 \qquad P(\theta \mid \mathcal{D}) \propto P(\mathcal{D} \mid \theta)$$

Bayesian Learning for Thumbtacks

$$P(\theta \mid \mathcal{D}) \;\; \propto \;\; P(\mathcal{D} \mid \theta)P(\theta)$$

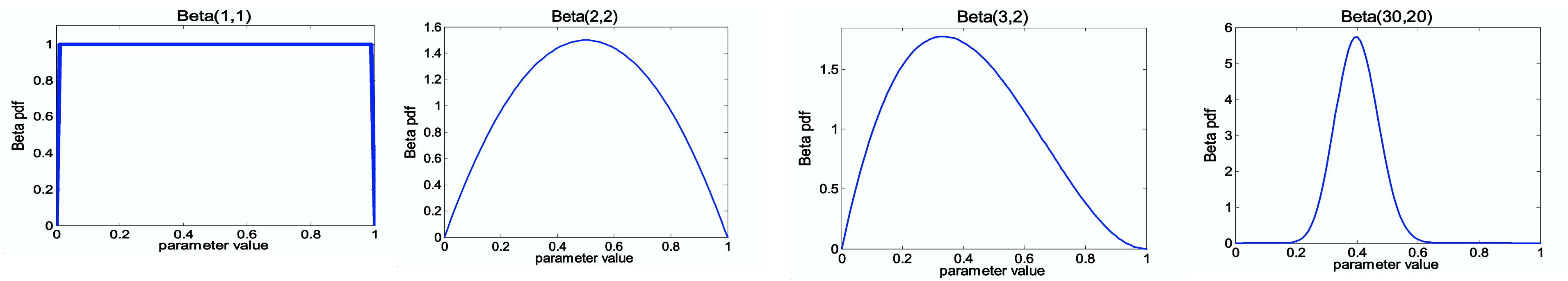
Likelihood: $P(\mathcal{D} \mid \theta) = \theta^{\alpha_H}(1-\theta)^{\alpha_T}$

- What should the prior be?
  - Represent expert knowledge
  - Simple posterior form
- For binary variables, commonly used prior is the Beta distribution:

$$P(\theta) = \frac{\theta^{\beta_H - 1}(1-\theta)^{\beta_T - 1}}{B(\beta_H, \beta_T)} \sim Beta(\beta_H, \beta_T)$$

Beta prior distribution – P(θ)

$$P(\theta) = \frac{\theta^{\beta_H - 1}(1-\theta)^{\beta_T - 1}}{B(\beta_H, \beta_T)} \sim Beta(\beta_H, \beta_T)$$



- Since the Beta distribution is *conjugate* to the Bernoulli distribution, the posterior distribution has a particularly simple form:

$$\begin{aligned} P(\theta \mid \mathcal{D}) &\propto P(\mathcal{D} \mid \theta) P(\theta) \\ &\propto \theta^{\alpha_H}(1-\theta)^{\alpha_T}\ \theta^{\beta_H - 1}(1-\theta)^{\beta_T - 1} \\ &= \theta^{\alpha_H + \beta_H - 1}(1-\theta)^{\alpha_T + \beta_T - 1} \\ &= Beta(\alpha_H + \beta_H, \alpha_T + \beta_T) \end{aligned}$$

Using Bayesian inference for prediction

- We now have a **distribution** over parameters
- For any specific *f*, a function of interest, compute the expected value of *f*:

$$E[f(\theta)] = \int_0^1 f(\theta) P(\theta \mid \mathcal{D}) d\theta$$

  - Integral is often hard to compute
  - *As more data is observed, posterior is more concentrated*
- MAP (Maximum a posteriori approximation): use most likely parameter to approximate the expectation

$$\widehat{\theta} = \arg\max_{\theta} P(\theta \mid \mathcal{D})$$

$$E[f(\theta)] \approx f(\widehat{\theta})$$

Outline of lectures

- Review of probability
- Maximum likelihood estimation

2 examples of Bayesian classifiers:

- **Naïve Bayes**
- Logistic regression

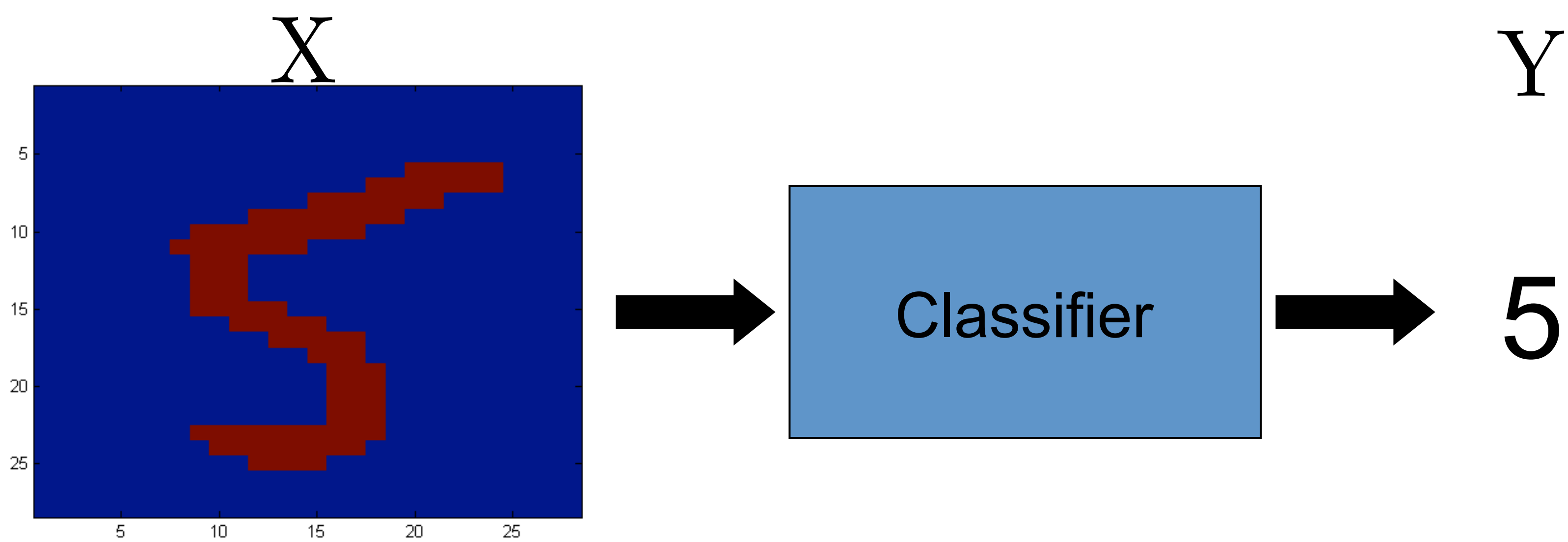
Bayesian Classification

- Problem statement:
  - Given features $X_1, X_2, \ldots, X_n$
  - Predict a label Y

[Next several slides adapted from:
Vibhav Gogate, Jonathan Huang, Luke Zettlemoyer, Carlos Guestrin, and Dan Weld]

Example Application

- **Digit Recognition**



- **$X_1,\ldots,X_n \in \{0,1\}$ (Black vs. White pixels)**
- **$Y \in \{0,1,2,3,4,5,6,7,8,9\}$**

The Bayes Classifier

- If we had the joint distribution on $\mathbf{X_1,\ldots,X_n}$ and **Y**, could predict using:

$$\arg\max_{Y} P(Y|X_1,\ldots,X_n)$$

  - (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

- 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:

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

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

- To classify, we'll simply compute these probabilities, one per class, and predict based on which one is largest

Model Parameters

- How many parameters are required to specify the likelihood, $P(X_1,\ldots,X_n|Y)$?
  - (Supposing that each image is 30x30 pixels)

- 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)

Naïve Bayes

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

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

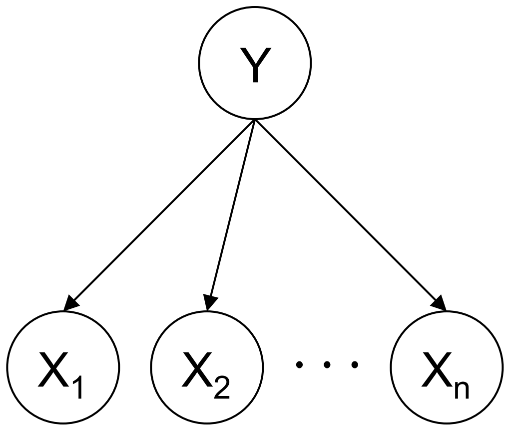
  - More generally:

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

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

The Naïve Bayes Classifier

- Given:
  - Prior P(Y)
  - *n* conditionally independent features $X_1, \ldots, X_1$, given the class Y
  - For each feature *i*, we specify $P(X_i|Y)$



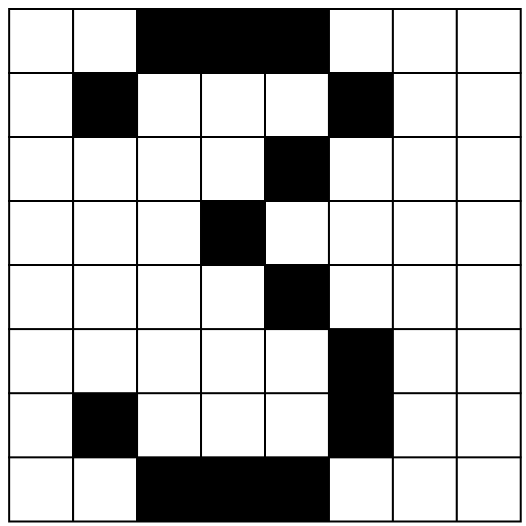
- Classification decision rule:

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

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

A Digit Recognizer

- Input: pixel grids



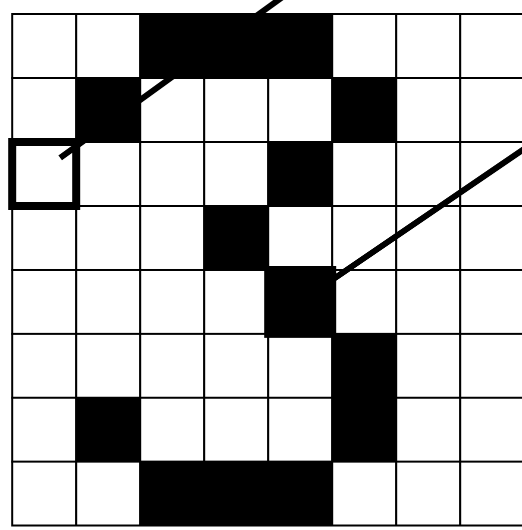
- Output: a digit 0-9

Are the naïve Bayes assumptions realistic here?

What has to be learned?

$P(Y)$

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:

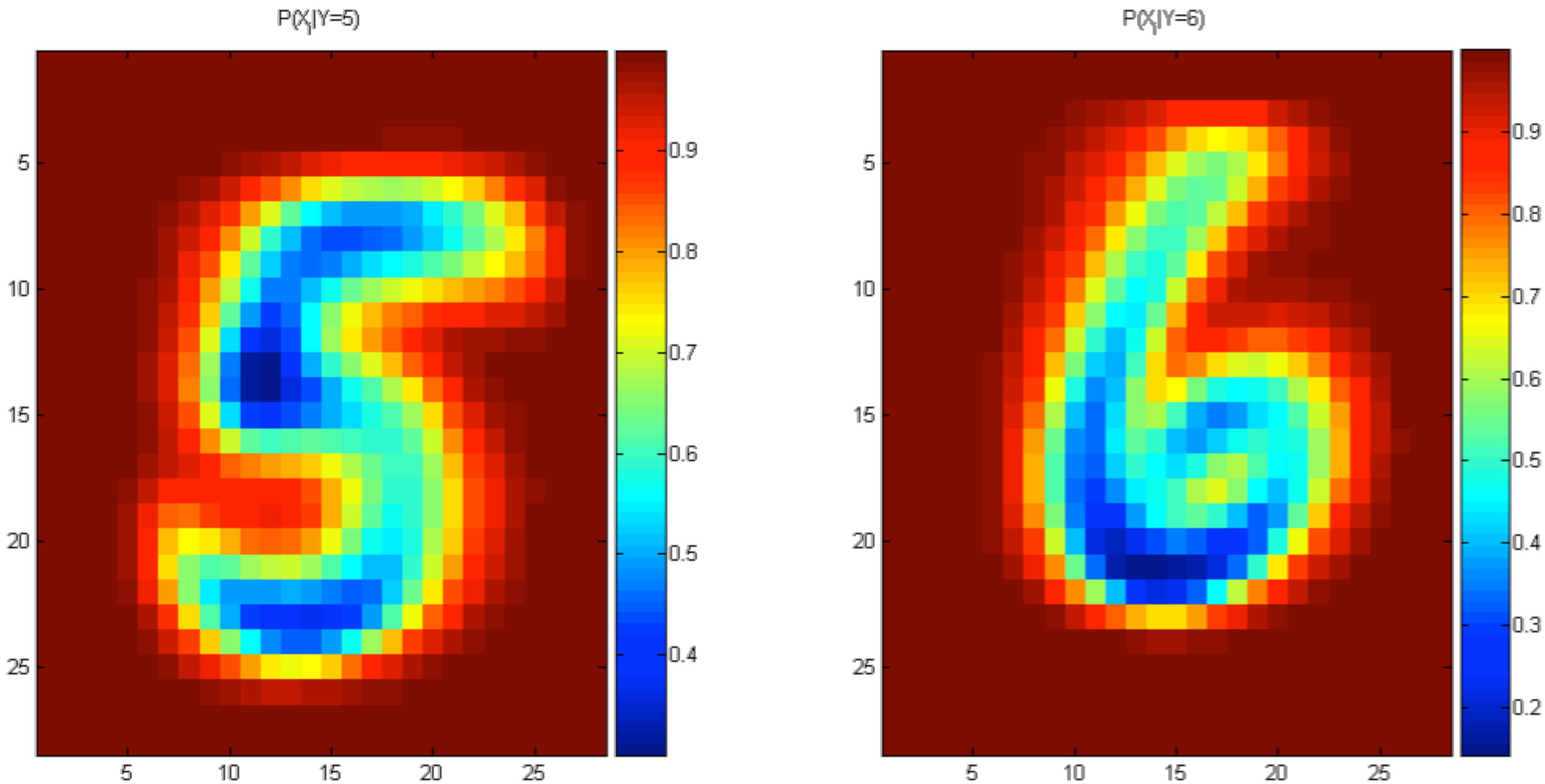
$$P(Y=y) = \frac{Count(Y=y)}{\sum_{y'} Count(Y=y')}$$

  - Observation distribution:

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

MLE for the parameters of NB

- Training amounts to, for each of the classes, averaging all of the examples together:



MAP estimation for NB

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

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

  - Observation distribution:

$$P(X_i = x | Y = y) = \frac{Count(X_i = x, Y = y) \textbf{ + a}}{\sum_{x'} Count(X_i = x', Y = y) \textbf{ + |X\_i|*a}}$$

- Called “smoothing”. Corresponds to Dirichlet prior!