

Bayesian methods & Naïve Bayes

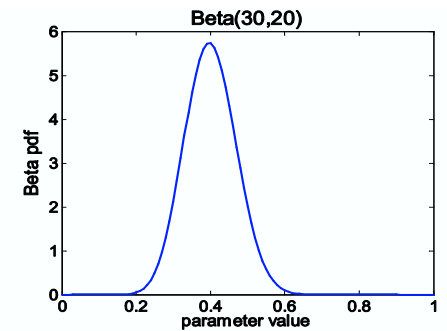
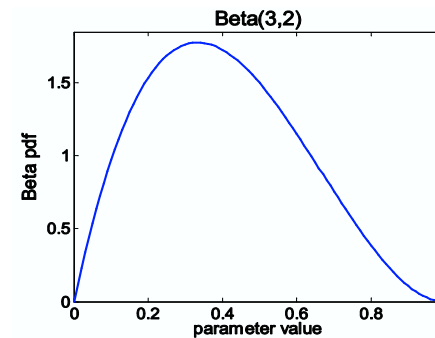
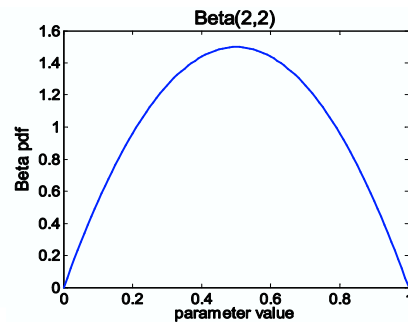
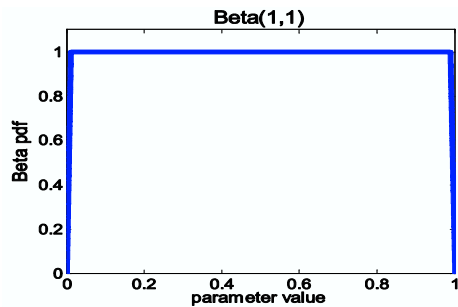
Lecture 18

David Sontag
New York University

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

Beta prior distribution – $P(\theta)$

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



- The posterior distribution:

$$P(\theta | \mathcal{D}) \propto P(\mathcal{D} | \theta)P(\theta)$$

$$\propto \theta^{\alpha_H} (1 - \theta)^{\alpha_T} \theta^{\beta_H-1} (1 - \theta)^{\beta_T-1}$$

$$= \theta^{\alpha_H + \beta_H - 1} (1 - \theta)^{\alpha_T + \beta_T - 1}$$

$$= \text{Beta}(\alpha_H + \beta_H, \alpha_T + \beta_T)$$

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 :

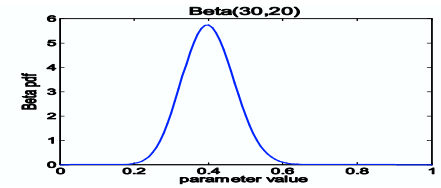
$$E[f(\theta)] = \int_0^1 f(\theta) P(\theta | \mathcal{D}) d\theta$$

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

$$\hat{\theta} = \arg \max_{\theta} P(\theta | \mathcal{D})$$

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

MAP for Beta distribution



$$P(\theta | \mathcal{D}) = \frac{\theta^{\beta_H + \alpha_H - 1} (1 - \theta)^{\beta_T + \alpha_T - 1}}{B(\beta_H + \alpha_H, \beta_T + \alpha_T)} \sim \text{Beta}(\beta_H + \alpha_H, \beta_T + \alpha_T)$$

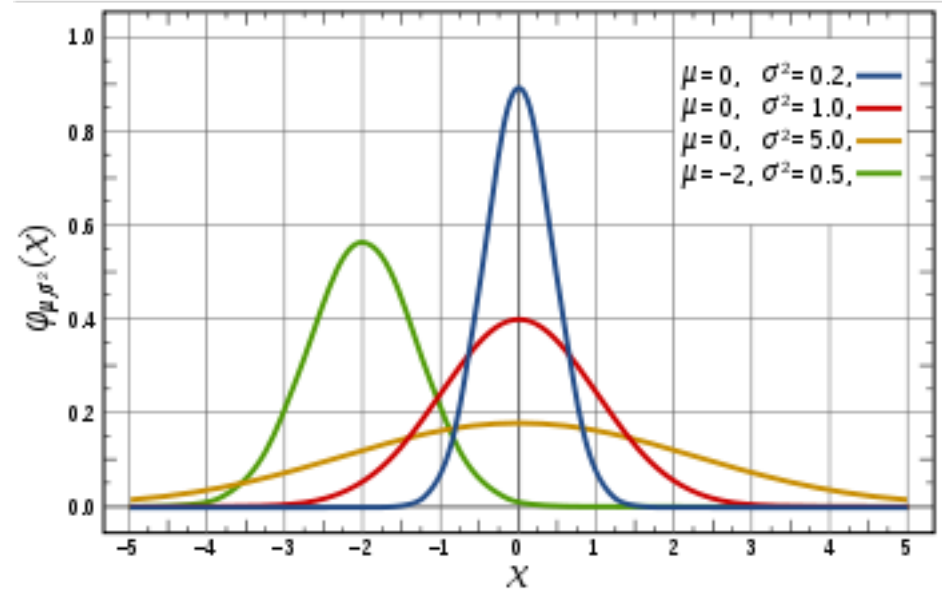
- MAP: use most likely parameter:

$$\hat{\theta} = \arg \max_{\theta} P(\theta | \mathcal{D}) = \frac{\alpha_H + \beta_H - 1}{\alpha_H + \beta_H + \alpha_T + \beta_T - 2}$$

- Beta prior equivalent to extra thumbtack flips
- As $N \rightarrow \infty$, prior is “forgotten”
- **But, for small sample size, prior is important!**

What about continuous variables?

- Billionaire says: If I am measuring a continuous variable, what can you do for me?
- You say: Let me tell you about Gaussians...



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

Some properties of Gaussians

- Affine transformation (multiplying by scalar and adding a constant) are Gaussian

- $X \sim N(\mu, \sigma^2)$

- $Y = aX + b \rightarrow Y \sim N(a\mu + b, a^2\sigma^2)$

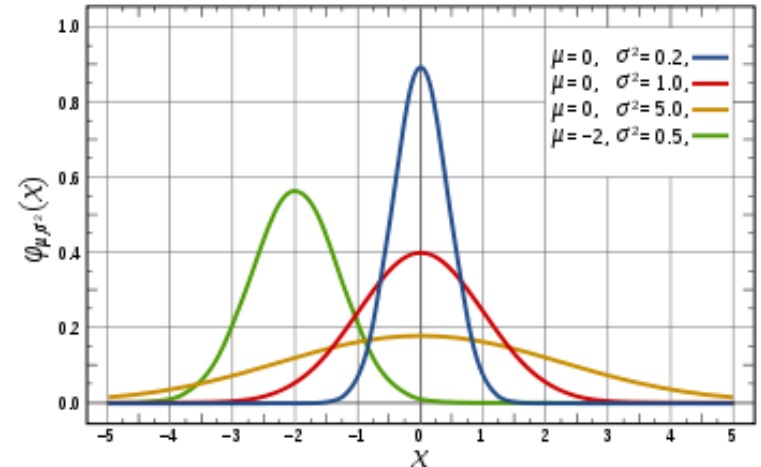
- Sum of Gaussians is Gaussian

- $X \sim N(\mu_X, \sigma_X^2)$

- $Y \sim N(\mu_Y, \sigma_Y^2)$

- $Z = X + Y \rightarrow Z \sim N(\mu_X + \mu_Y, \sigma_X^2 + \sigma_Y^2)$

- Easy to differentiate, as we will see soon!



Learning a Gaussian

- Collect a bunch of data
 - Hopefully, i.i.d. samples
 - e.g., exam scores
- Learn parameters
 - Mean: μ
 - Variance: σ

x_i $i =$	Exam Score
0	85
1	95
2	100
3	12
...	...
99	89

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

MLE for Gaussian: $P(x | \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

- Prob. of i.i.d. samples $D=\{x_1, \dots, x_N\}$:

$$P(\mathcal{D} | \mu, \sigma) = \left(\frac{1}{\sigma\sqrt{2\pi}} \right)^N \prod_{i=1}^N e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

$$\mu_{MLE}, \sigma_{MLE} = \arg \max_{\mu, \sigma} P(\mathcal{D} | \mu, \sigma)$$

- Log-likelihood of data:

$$\begin{aligned} \ln P(\mathcal{D} | \mu, \sigma) &= \ln \left[\left(\frac{1}{\sigma\sqrt{2\pi}} \right)^N \prod_{i=1}^N e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} \right] \\ &= -N \ln \sigma\sqrt{2\pi} - \sum_{i=1}^N \frac{(x_i - \mu)^2}{2\sigma^2} \end{aligned}$$

Your second learning algorithm: MLE for mean of a Gaussian

- What's MLE for mean?

$$\begin{aligned}\frac{d}{d\mu} \ln P(\mathcal{D} \mid \mu, \sigma) &= \frac{d}{d\mu} \left[-N \ln \sigma \sqrt{2\pi} - \sum_{i=1}^N \frac{(x_i - \mu)^2}{2\sigma^2} \right] \\ &= \frac{d}{d\mu} \left[-N \ln \sigma \sqrt{2\pi} \right] - \sum_{i=1}^N \frac{d}{d\mu} \left[\frac{(x_i - \mu)^2}{2\sigma^2} \right] \\ &= \sum_{i=1}^N \frac{(x_i - \mu)}{\sigma^2} = 0 \\ &= \sum_{i=1}^N x_i - N\mu = 0\end{aligned}$$

$$\hat{\mu}_{MLE} = \frac{1}{N} \sum_{i=1}^N x_i$$

MLE for variance

- Again, set derivative to zero:

$$\begin{aligned}\frac{d}{d\sigma} \ln P(\mathcal{D} \mid \mu, \sigma) &= \frac{d}{d\sigma} \left[-N \ln \sigma \sqrt{2\pi} - \sum_{i=1}^N \frac{(x_i - \mu)^2}{2\sigma^2} \right] \\ &= \frac{d}{d\sigma} \left[-N \ln \sigma \sqrt{2\pi} \right] - \sum_{i=1}^N \frac{d}{d\sigma} \left[\frac{(x_i - \mu)^2}{2\sigma^2} \right] \\ &= -\frac{N}{\sigma} + \sum_{i=1}^N \frac{(x_i - \mu)^2}{\sigma^3} = 0\end{aligned}$$

$$\hat{\sigma}_{MLE}^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \hat{\mu})^2$$

Learning Gaussian parameters

- MLE:

$$\hat{\mu}_{MLE} = \frac{1}{N} \sum_{i=1}^N x_i$$

$$\hat{\sigma}_{MLE}^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \hat{\mu})^2$$

- MLE for the variance of a Gaussian is **biased**

- Expected result of estimation is **not** true parameter!
- Unbiased variance estimator:

$$\hat{\sigma}_{unbiased}^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \hat{\mu})^2$$

Bayesian learning of Gaussian parameters

- Conjugate priors
 - Mean: Gaussian prior
 - Variance: Wishart Distribution

- Prior for mean:

$$P(\mu | \eta, \lambda) = \frac{1}{\lambda\sqrt{2\pi}} e^{-\frac{(\mu-\eta)^2}{2\lambda^2}}$$

Naïve Bayes

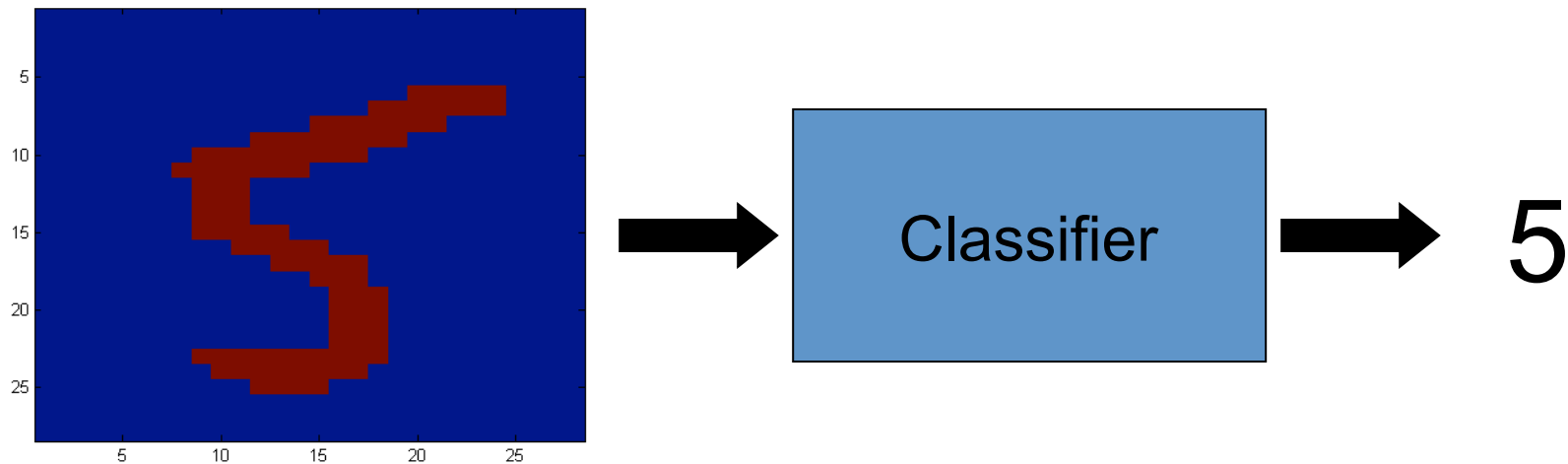
Slides adapted from Vibhav Gogate, Jonathan Huang,
Luke Zettlemoyer, Carlos Guestrin, and Dan Weld

Bayesian Classification

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

Example Application

- **Digit Recognition**



- $X_1, \dots, X_n \in \{0,1\}$ (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 $\mathbf{X}_1, \dots, \mathbf{X}_n$ and \mathbf{Y} , could predict using:

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

- (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!

$$P(Y|X_1, \dots, X_n) = \frac{\overset{\text{Likelihood}}{P(X_1, \dots, X_n|Y)} \overset{\text{Prior}}{P(Y)}}{\underset{\text{Normalization Constant}}{P(X_1, \dots, X_n)}}$$

- 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, $P(X_1, \dots, X_n | Y)$?
 - (Supposing that each image is 30x30 pixels)
- The problem with explicitly modeling $P(X_1, \dots, X_n | Y)$ 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:

$$\begin{aligned}P(X_1, X_2|Y) &= P(X_1|X_2, Y)P(X_2|Y) \\ &= P(X_1|Y)P(X_2|Y)\end{aligned}$$

- More generally:

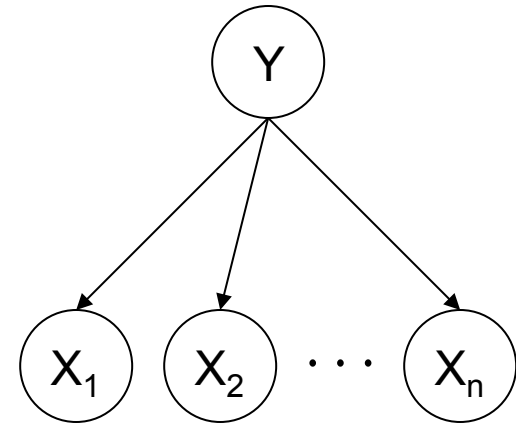
$$P(X_1 \dots X_n|Y) = \prod_i P(X_i|Y)$$

- How many parameters now?
 - Suppose \mathbf{X} is composed of n binary features

The Naïve Bayes Classifier

- Given:

- Prior $P(Y)$
- n conditionally independent features \mathbf{X} given the class Y
- For each X_i , we have likelihood $P(X_i | Y)$



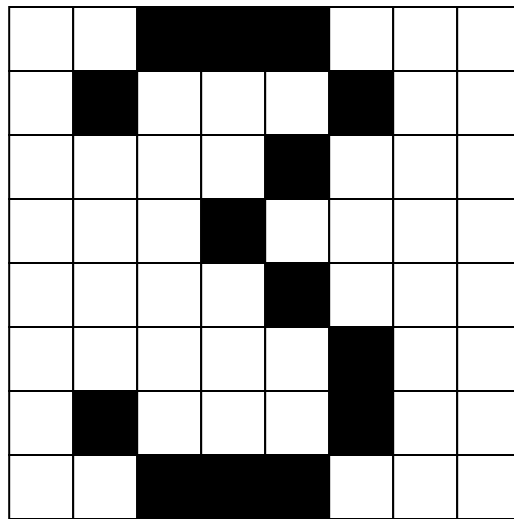
- Decision rule:

$$\begin{aligned} y^* = h_{NB}(\mathbf{x}) &= \arg \max_y P(y) P(x_1, \dots, x_n | y) \\ &= \arg \max_y P(y) \prod_i P(x_i | y) \end{aligned}$$

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 $\langle i,j \rangle$
- 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.

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

- Here: lots of features, each is binary valued

- Naïve Bayes model:

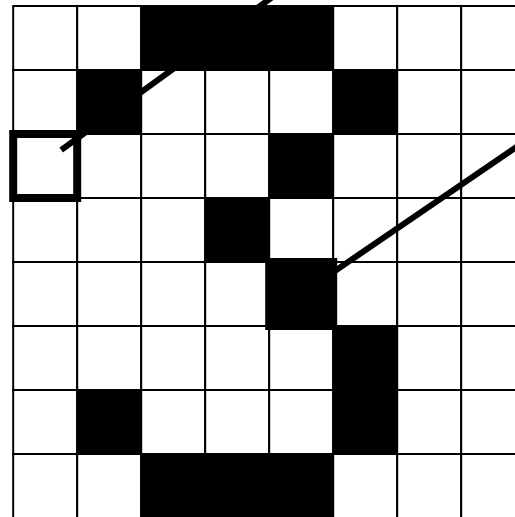
$$P(Y|F_{0,0} \dots F_{15,15}) \propto P(Y) \prod_{i,j} P(F_{i,j}|Y)$$

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

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



$P(F_{3,1} = on|Y)$ $P(F_{5,5} = on|Y)$

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

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

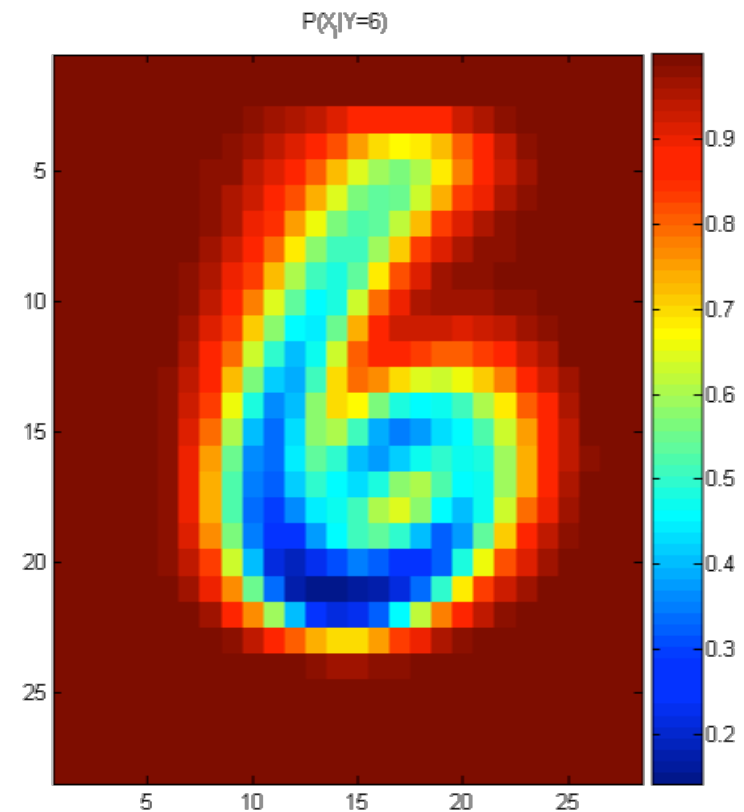
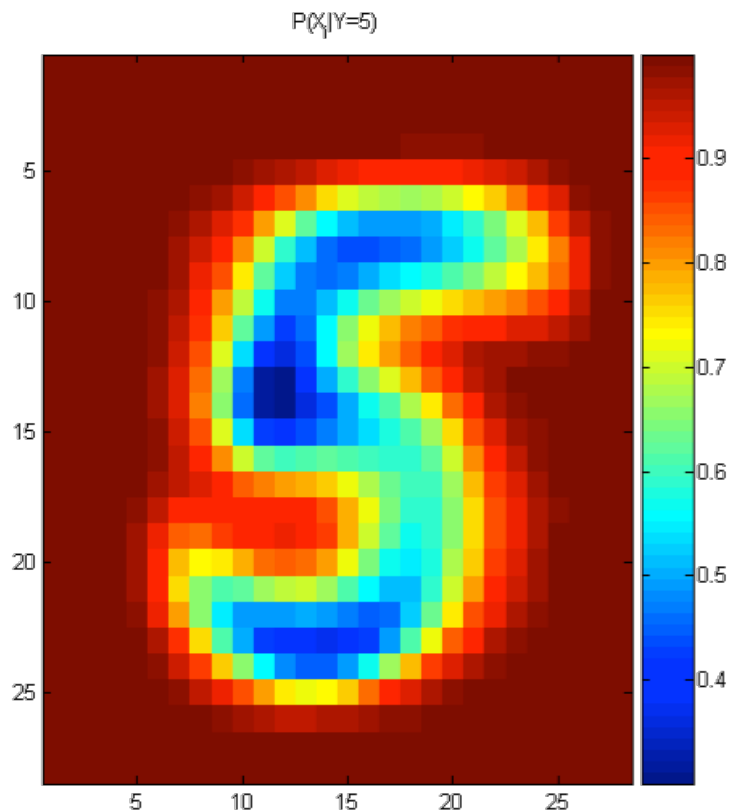
$$P(Y = y) = \frac{\text{Count}(Y = y)}{\sum_{y'} \text{Count}(Y = y')}$$

- Observation distribution:

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

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

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

- Observation distribution:

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

- Called “smoothing”. Corresponds to Dirichlet prior!