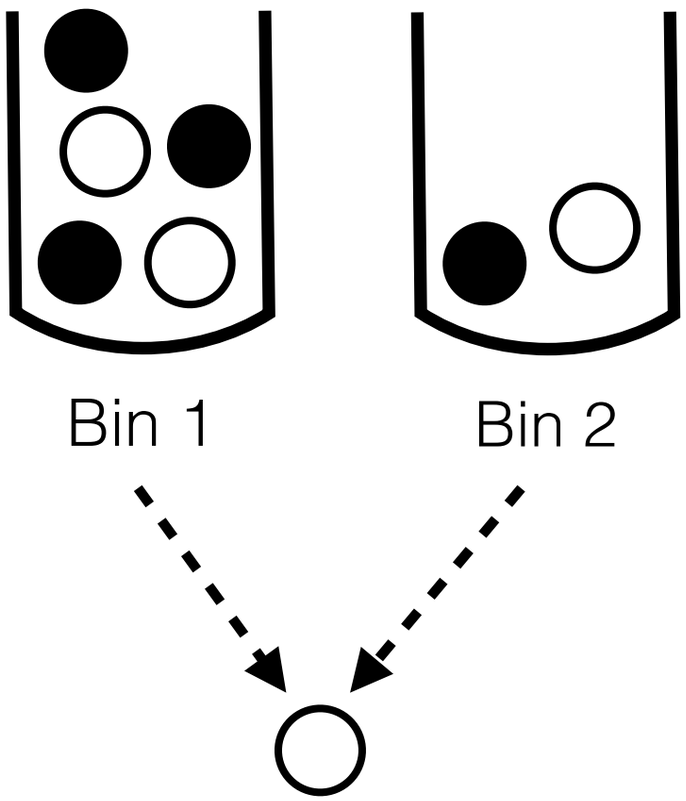
Lecture 21

Inference

CS70 Spring 2015

Inference: Bayes' rule

- Recall simple balls and bins problem from Note 14



$$\Pr[\text{Bin } 1] = \Pr[\text{Bin } 2] = \frac{1}{2}$$

$$\Pr[\text{Bin } 1 \mid W] = \frac{\Pr[W \mid \text{Bin } 1] \cdot \Pr[\text{Bin } 1]}{\Pr[W]}$$

$$= \frac{\frac{2}{5} \times \frac{1}{2}}{\frac{2}{5} \times \frac{1}{2} + \frac{1}{2} \times \frac{1}{2}} = \frac{4}{9}$$

$$\Pr[\text{Bin } 2 \mid W] = 1 - \Pr[\text{Bin } 1 \mid W] = \frac{5}{9}$$

- So white ball more likely to have come from Bin 2

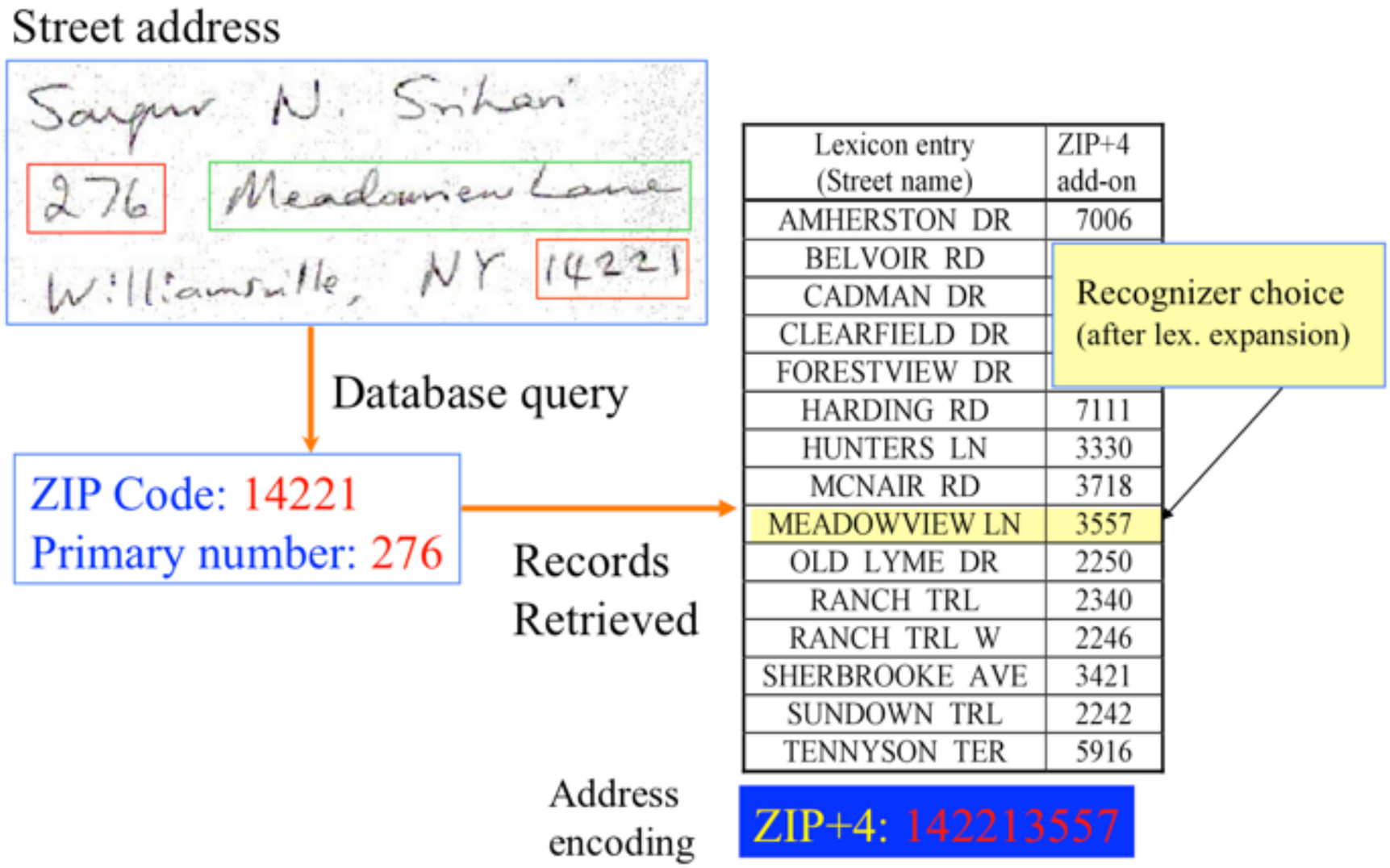
Handwriting recognition

- Money detection in check deposit

→ $ 2.56

Handwriting recognition

- Address detection in postal mail



(S. N. Srihari and E. J. Keubert, "Integration of handwritten address interpretation technology into the United States Postal Service Remote Computer Reader System", ICDAR 1997)

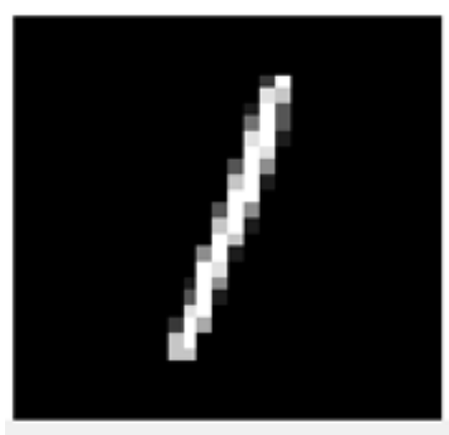
Digit classification

- Given an image, determine what digit is in the image
- MNIST dataset: each image is 28 x 28 grayscale

http://yann.lecun.com/exdb/mnist/

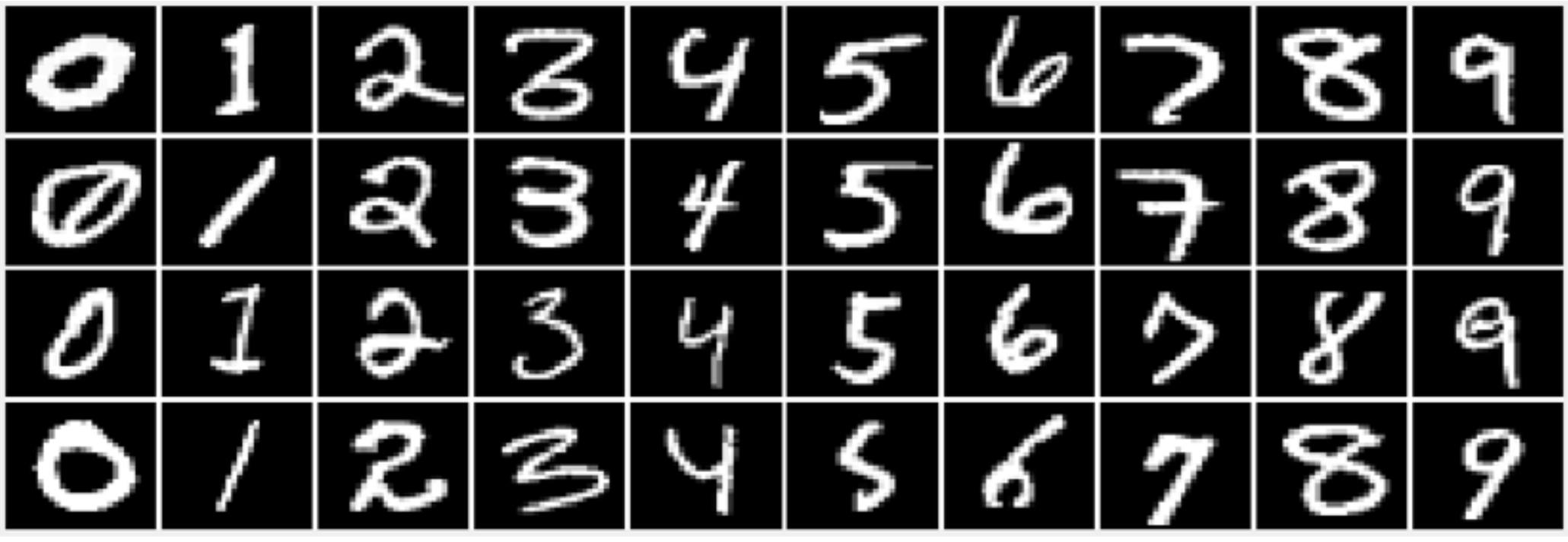
Modeling the digits

- If we know how the digits are generated, then we know how to classify them
  - ✳ For each digit j, have model M(j) that generates all possible images of digit j
  - ✳ Given an image x, check which model M(j) can generate x
  - ✳ If our model is perfect, there is only one such digit for every x
  - ✳ Classification: check which model M(j) is consistent with x



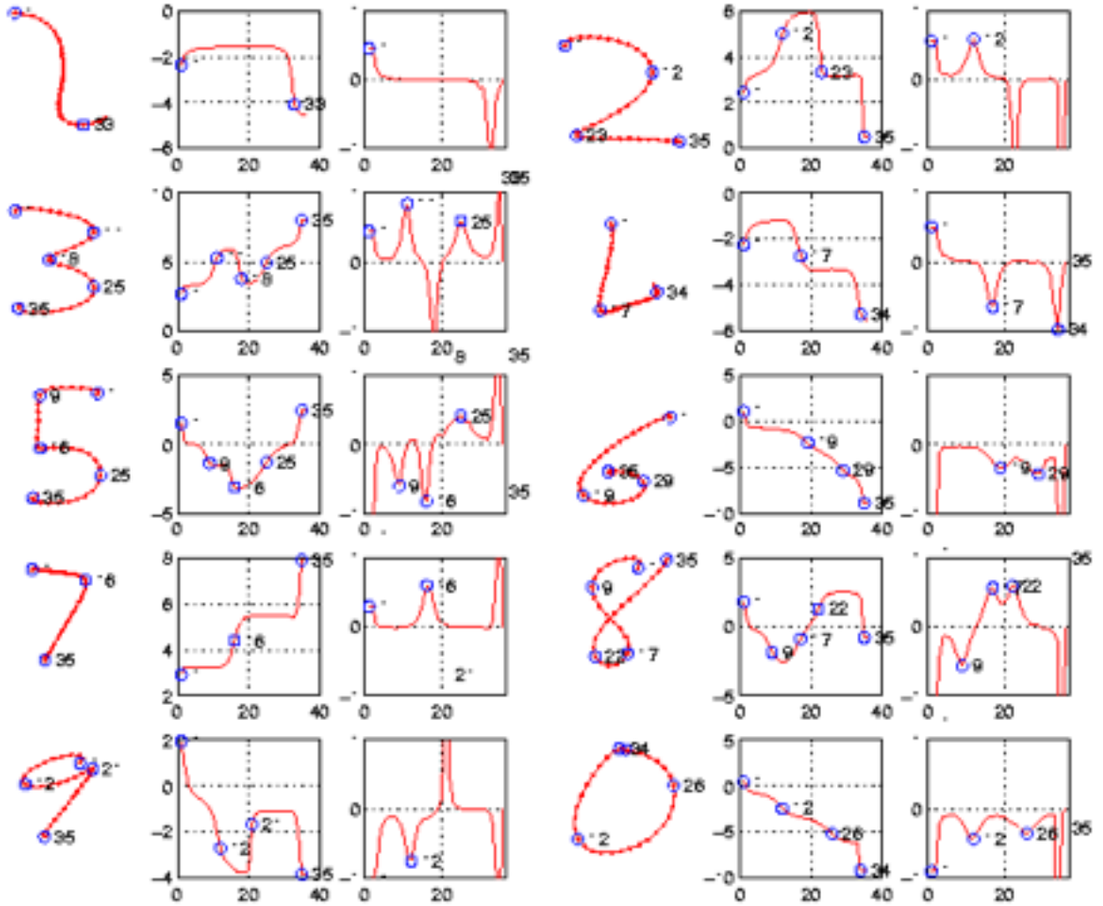
Modeling the digits

- Try to build a model for each digit by:
  - ✳ Simple rules for writing digits: “0” is a simple loop, “1” has only one stroke, “8” has 2 loops, …
    - Not always satisfied, need many exceptions or more complicated rules



Modeling the digits

- Try to build a model for each digit by:
  - ✳ Simple rules for writing digits: “0” is a simple loop, “1” has only one stroke, “8” has 2 loops, …
  - ✳ Considering curvatures of the digits (how the angles change)
    - Works well, but complicated
  - ✳ Can we get away with something simpler?



(M. Kamvysselis, “Digit recognition in curvature space”, 1999)

Probabilistic approach

- Treat variations among images of a digit as a probability distribution over **all** the images x
  - ✳ Distribution $P_j(x)$ generates images x from digit j, but by (small) random chance can look like other digits
  - ✳ Imperfect model, represents our uncertainty/ambiguity, but Bayes' rule to the rescue!

$P_1(x) =$  $\cdots$

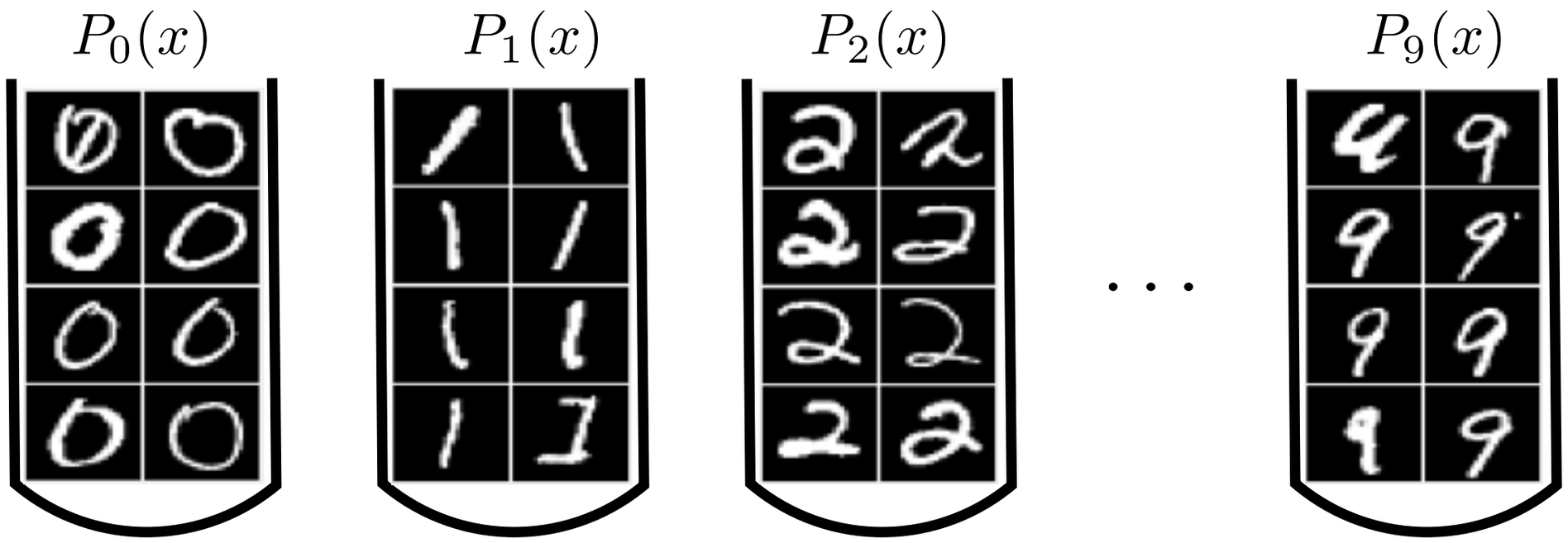
$P_2(x) =$  $\cdots$

$P_8(x) =$  $\cdots$

Back to balls and bins

- We have 10 bins, each corresponds to a digit
- Prior probability $\pi_j = \Pr[y = j]$
- In bin j, have conditional distribution $P_j(x) = P(x \mid y = j)$
- Overall distribution of image (ball) x is mixture distribution

$$P(x) = \pi_0\, P_0(x) + \pi_1\, P_1(x) + \pi_2\, P_2(x) + \cdots + \pi_9\, P_9(x)$$

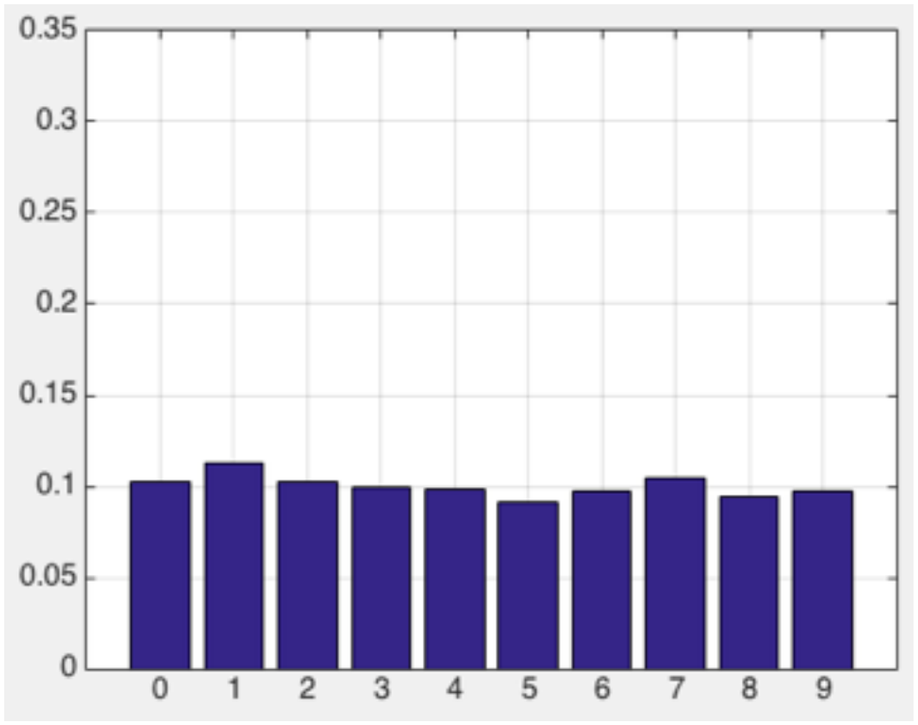


Back to balls and bins

- Suppose we are just given a ball (image) x
- Which bin (digit) y did it come from?
- We know how to solve this! **Bayes' rule:**

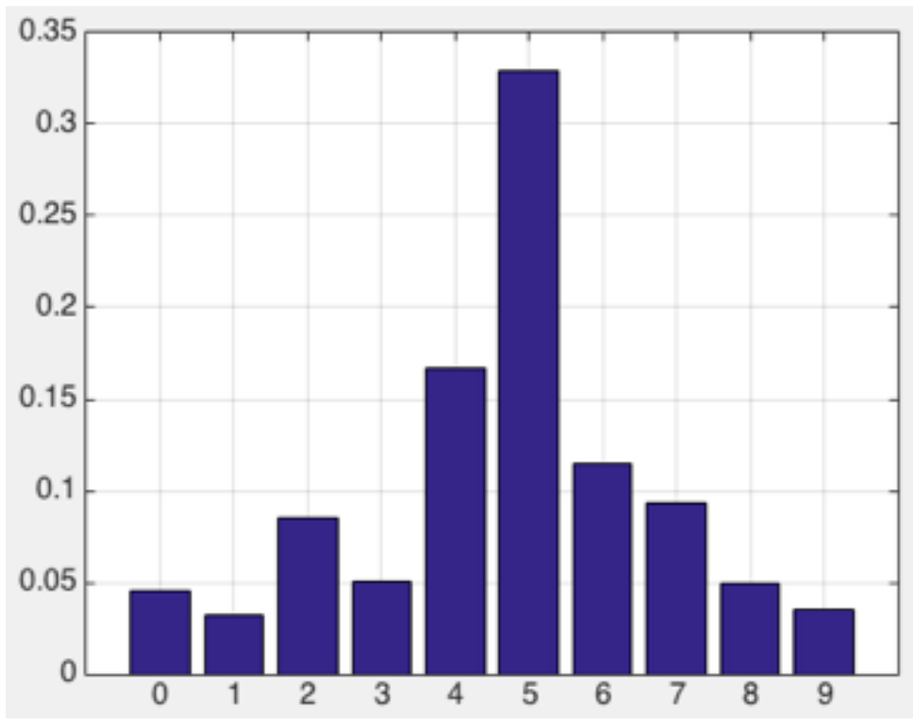
$$\Pr[y = j \mid x] = \frac{P(x \mid y = j) \cdot \Pr[y = j]}{P(x)} = \frac{\pi_j \, P_j(x)}{\sum_i \pi_i \, P_i(x)}$$

prior

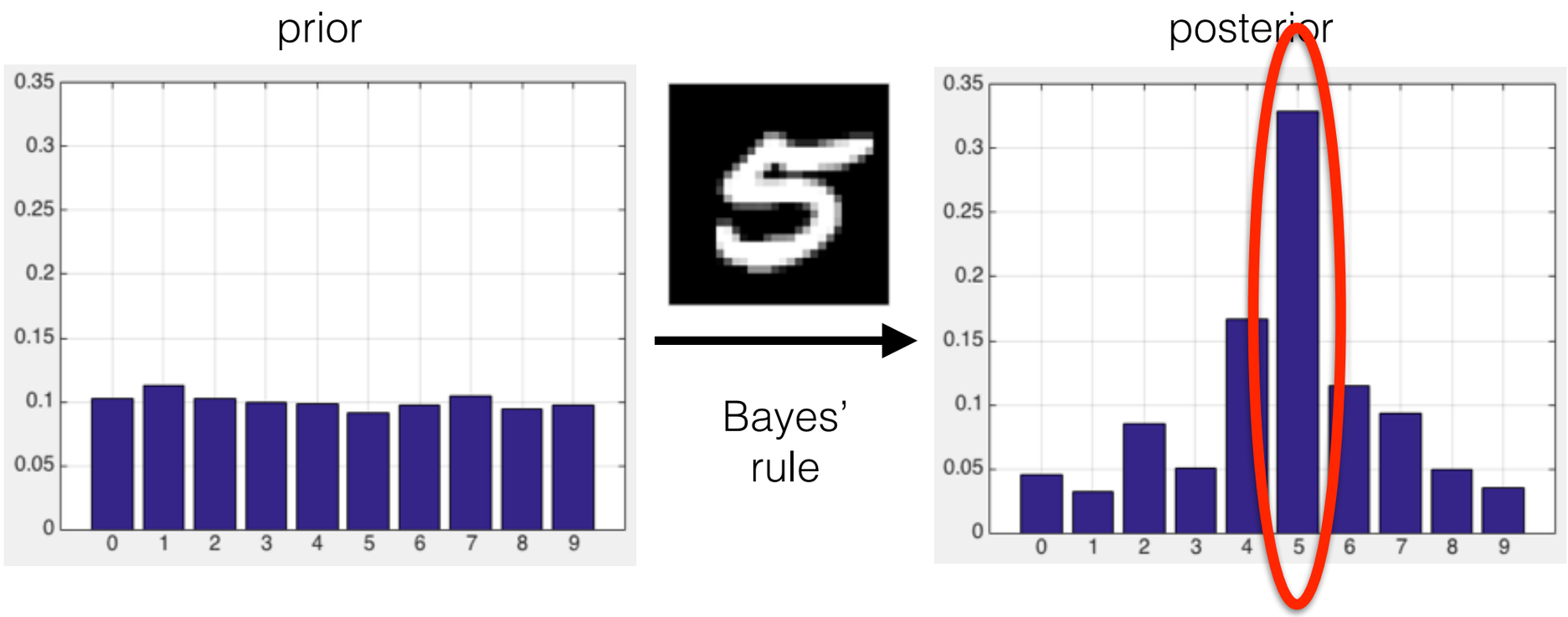


Bayes' rule

posterior



Classification as inference



- Which is the most likely bin from which x was drawn?
- The one that maximizes the posterior probability!

$$h^*(x) = \arg\max_j \Pr[y = j \mid x]$$

- This is the *Bayes-optimal* estimator. Also called the *maximum a posteriori (MAP)* estimator.

Classification as inference

- Classify image x to the class that maximizes posterior:

$$h^*(x) = \arg\max_j \Pr[y = j \mid x]$$

- Denominator P(x) is independent of j:

$$\Pr[y = j \mid x] = \frac{P(x \mid y = j) \cdot \Pr[y = j]}{P(x)} = \frac{\pi_j \, P_j(x)}{\sum_i \pi_i \, P_i(x)}$$

- So find MAP estimator by maximizing {prior} x {likelihood}:

$$h^*(x) = \arg\max_j \pi_j \, P_j(x)$$

- If we know the prior and likelihood, then can classify!
- But where do they come from?

Estimating the distributions

- Use training data to estimate the prior $\pi_j = \Pr[y = j]$ and the class-conditional distributions $P_j(x) = P(x \mid y = j)$
  - ✳ MNIST dataset: 60K training data, 10K test data
- Estimating the π_j is easy:

$$\hat{\pi}_j = \frac{n_j}{n} = \frac{\#\text{ of examples of class } j}{\text{total }\#\text{ of examples}}$$

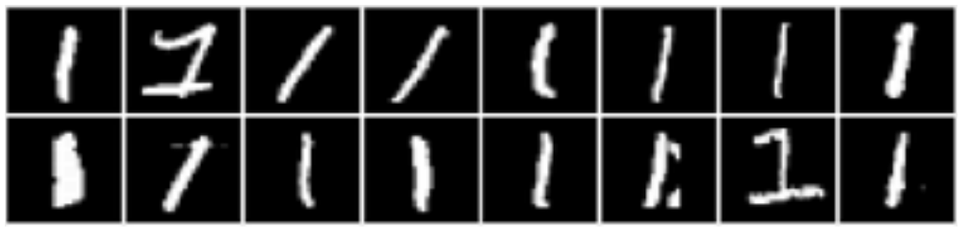
- From MNIST:

j	0	1	2	3	4	5	6	7	8	9
$\hat{\pi}_j$ (%)	9.87	11.24	9.93	10.22	9.74	9.03	9.86	10.44	9.75	9.92

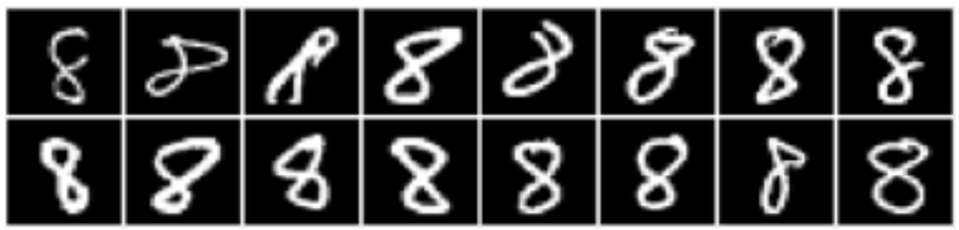
- But estimating the $P_j(x)$ is difficult!

Estimating class-conditional distributions

- How to estimate the conditional distributions $P_j(x)$?
  - ✳ Have training examples from each class, but what distribution to use?
  - ✳ Each example is a 28 x 28 grayscale image $\rightarrow x \in [0,1]^{784}$

$P_1(x) = \left\{ \right.$  $\left. \cdots \right\}$

$P_2(x) = \left\{ \right.$  $\left. \cdots \right\}$

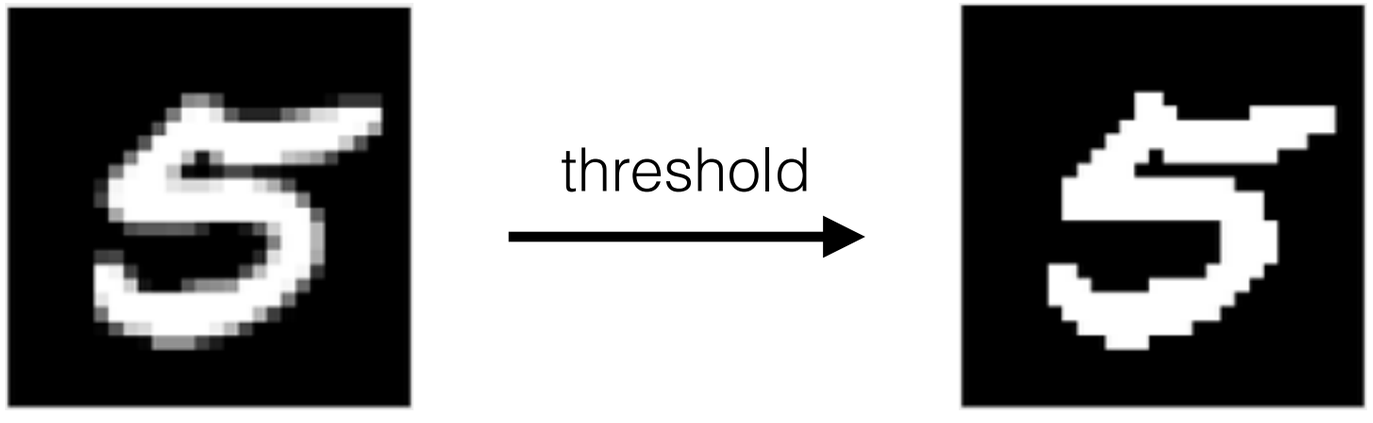
$P_8(x) = \left\{ \right.$  $\left. \cdots \right\}$

Estimating class-conditional distributions

- How to estimate the conditional distributions $P_j(x)$?
  - Have training examples from each class, but what distribution to use?
  - Each example is a 28 x 28 grayscale image $\rightarrow x \in [0,1]^{784}$
- **Idea:** Don't have to model $P_j(x)$ accurately
  - Can use a very simple, crude model, that is easy to learn
  - But still gives good performance, with the help of Bayes' rule
- We will see two simple models:
  - Naive Bayes: Assume $P_j(x)$ is product of independent coin flips
  - Gaussian model: Assume $P_j(x)$ is multivariate Gaussian

Naive Bayes

- Convert grayscale images to binary



$$\text{binary data}$$
$$x \in \{0,1\}^{784}$$

- A general distribution over $\{0,1\}^{784}$ has $2^{784}-1$ parameters
- Assume that **within each class**, the individual pixel values are independent:

$$P_j(x) = P_{j1}(x_1) \cdot P_{j2}(x_2) \cdots P_{j,784}(x_{784})$$

- Each P_{ji} is a coin flip: easy to estimate!
- Now only have 784 parameters to learn

Estimating bias of a coin

- Pick a class j and a pixel i. We use the biased coin flip

$$P_{ji}(x_i) = \begin{cases} p_{ji} & \text{if } x_i = 1 \\ 1 - p_{ji} & \text{if } x_i = 0 \end{cases}$$

- Can write as:

$$P_{ji}(x_i) = p_{ji}^{x_i}(1 - p_{ji})^{1-x_i}$$

- We need to estimate $p_{ji} = \Pr[x_i = 1 \mid y = j]$
- From training examples, let

$$n_j = \#\text{ of examples of class } j$$
$$n_{ji} = \#\text{ of examples of class } j \text{ with } x_i = 1$$

- Then our estimator is $\hat{p}_{ji} = \dfrac{n_{ji}}{n_j}$

Smoothed estimate of coin bias

- We estimate $p_{ji} = \Pr[x_i = 1 \mid y = j]$ via $\hat{p}_{ji} = \dfrac{n_{ji}}{n_j}$ where

$$n_j = \#\text{ of examples of class } j$$
$$n_{ji} = \#\text{ of examples of class } j \text{ with } x_i = 1$$

- This causes problems if $n_{ji} = 0$ or $n_{ji} = n_j$

  If all examples have $x_i = 1$, does that mean it is always 1?

  No, it can also happen by random chance.

- Instead, use "Laplace smoothing": $\hat{p}_{ji} = \dfrac{n_{ji} + 1}{n_j + 2}$
- This is as if we have 2 additional samples in class j: one with $x_i = 0$ and one with $x_i = 1$
- Can also interpret as effect of uniform prior on p_{ji}

Naive Bayes classifier

- Data space $\mathcal{X} = \{0,1\}^{784}$, label space $\mathcal{Y} = \{0,1,\ldots,9\}$
- From training data, estimate:
  - prior $\{\pi_j : 0 \leq j \leq 9\}$
  - conditionals $\{p_{ji} : 0 \leq j \leq 9, 1 \leq i \leq 784\}$
- Given an image $x = (x_1, \ldots, x_{784})$, conditional likelihood is

$$P_j(x) = \prod_{i=1}^{784} P_{ji}(x_i) = \prod_{i=1}^{784} p_{ji}^{x_i}(1-p_{ji})^{1-x_i}$$

- Then classify image x as

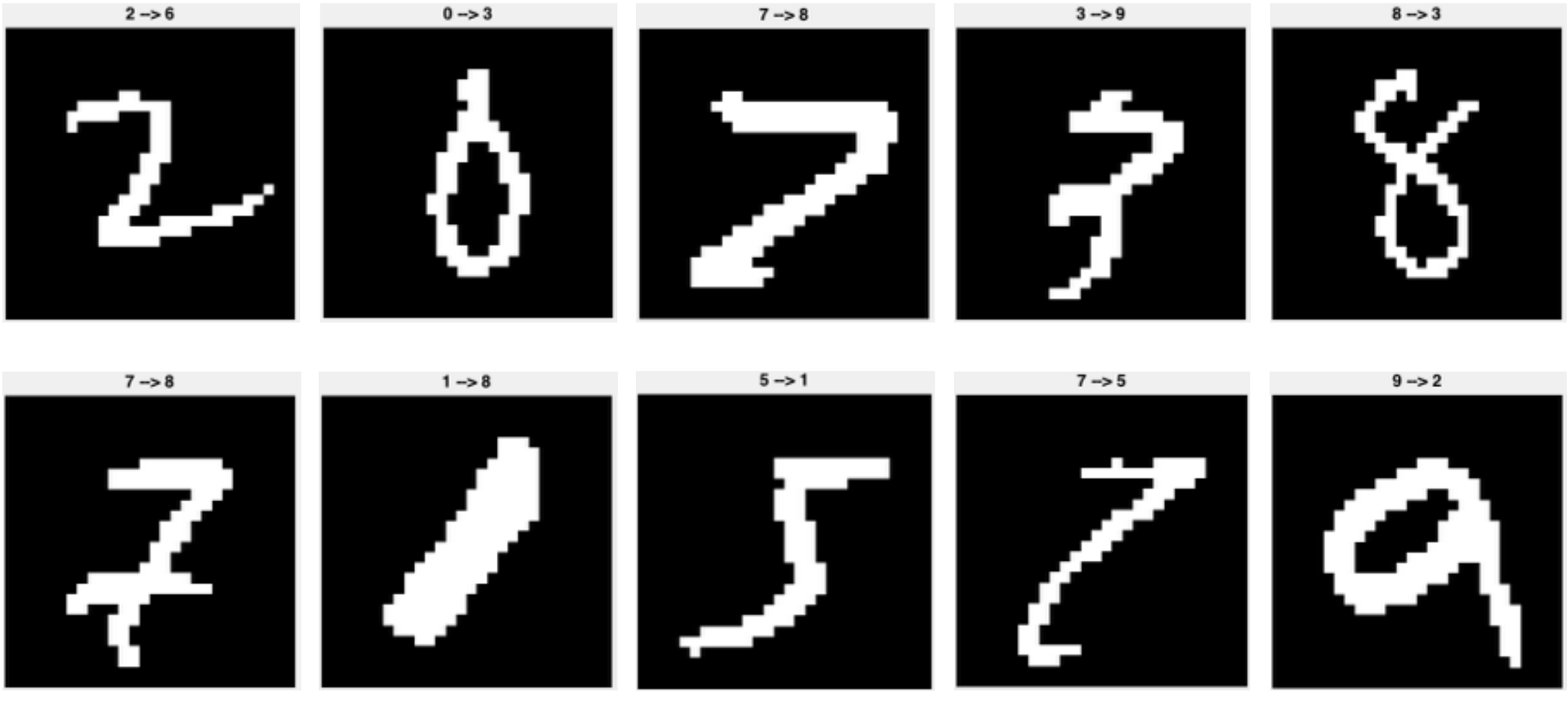
$$\arg\max_j \ \pi_j \prod_{i=1}^{784} p_{ji}^{x_i}(1-p_{ji})^{1-x_i}$$

- To avoid underflow, take the log:

$$\arg\max_j \ \log \pi_j + \sum_{i=1}^{784} \left(x_i \log p_{ji} + (1-x_i)\log(1-p_{ji}) \right)$$

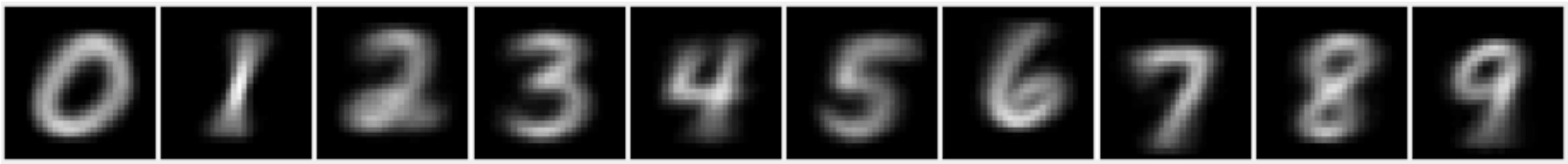
Naive Bayes on MNIST

- Error rate: **15.4%** (on 10K test data) —> pretty good!
- Some images that are misclassified:

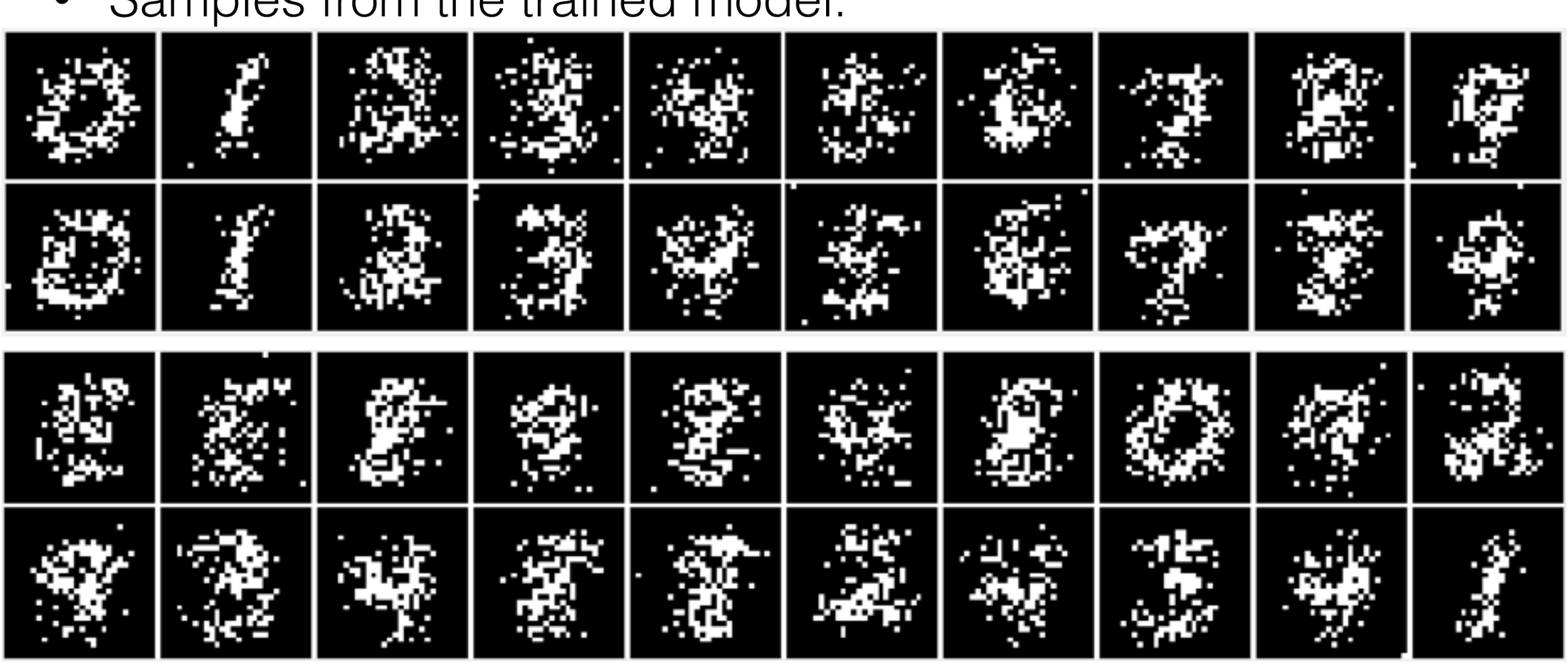


Naive Bayes on MNIST

- Error rate: **15.4%** (on 10K test data) —> pretty good!
- Mean vectors for each class (the p_{ji}'s):

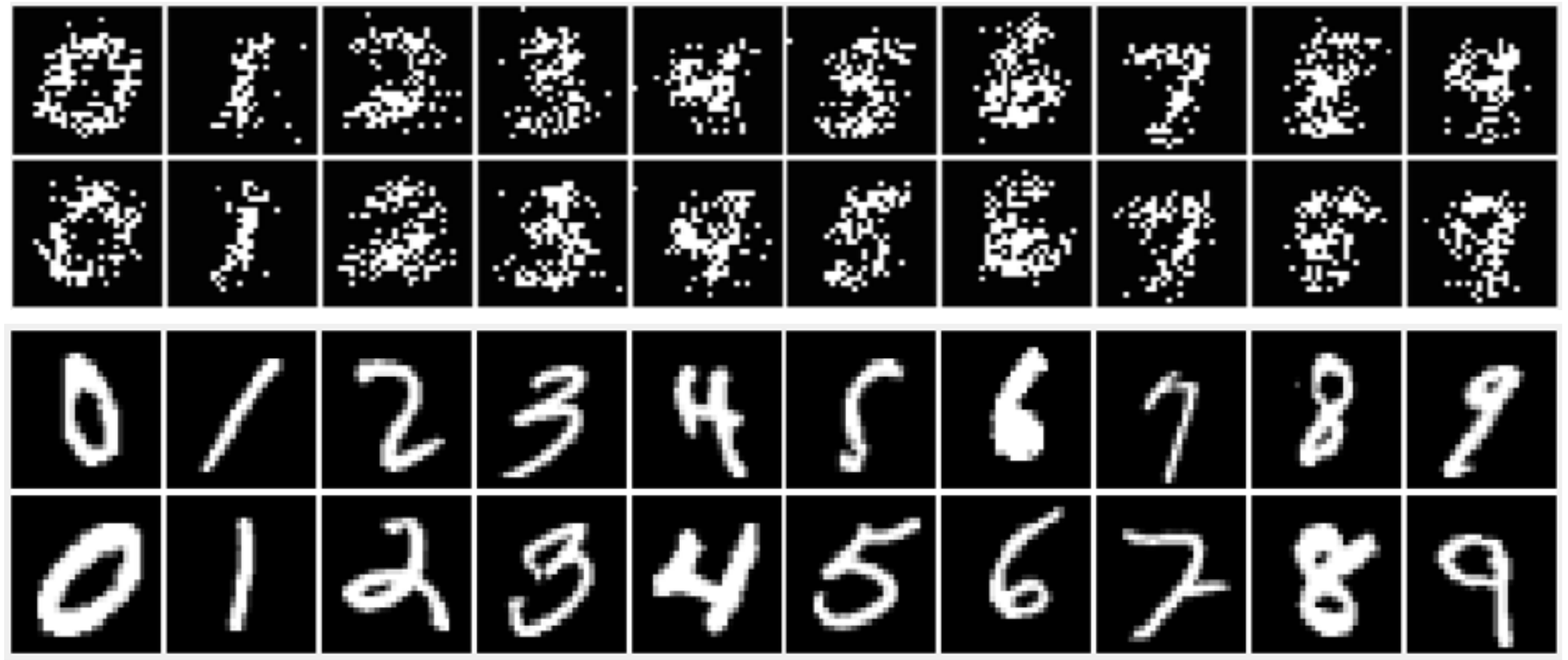


- Samples from the trained model:



Naive Bayes on MNIST

- Samples from Naive Bayes model look different from data:



- Naive Bayes is too simple, doesn't model the data well
  - ✳ Independence assumption is very not realistic
  - ✳ But good enough for our purposes, since only want MAP estimate
  - ✳ Trade-off: Model accuracy vs. complexity

Modeling correlation

- Independence assumption is too… naive
- Need to take into account correlation between pixel values

Given two random variables X_i, X_j with $\mathrm{E}(X_i) = \mu_i$, $\mathrm{E}(X_j) = \mu_j$

Covariance: $\mathrm{Cov}(X_i, X_j) = \mathrm{E}(X_i - \mu_i)(X_j - \mu_j)$

Variance: $\mathrm{Var}(X_i) = \mathrm{E}(X_i - \mu_i)^2 = \mathrm{Cov}(X_i, X_i)$

Summarize in covariance matrix Σ:

$$\Sigma = \begin{pmatrix} \Sigma_{ii} & \Sigma_{ij} \\ \Sigma_{ji} & \Sigma_{jj} \end{pmatrix} = \begin{pmatrix} \mathrm{Var}(X_i) & \mathrm{Cov}(X_i, X_j) \\ \mathrm{Cov}(X_j, X_i) & \mathrm{Var}(X_j) \end{pmatrix}$$

$$= \mathrm{E}\left[\begin{pmatrix} X_i - \mu_i \\ X_j - \mu_j \end{pmatrix} \begin{pmatrix} X_i - \mu_i & X_j - \mu_j \end{pmatrix} \right]$$

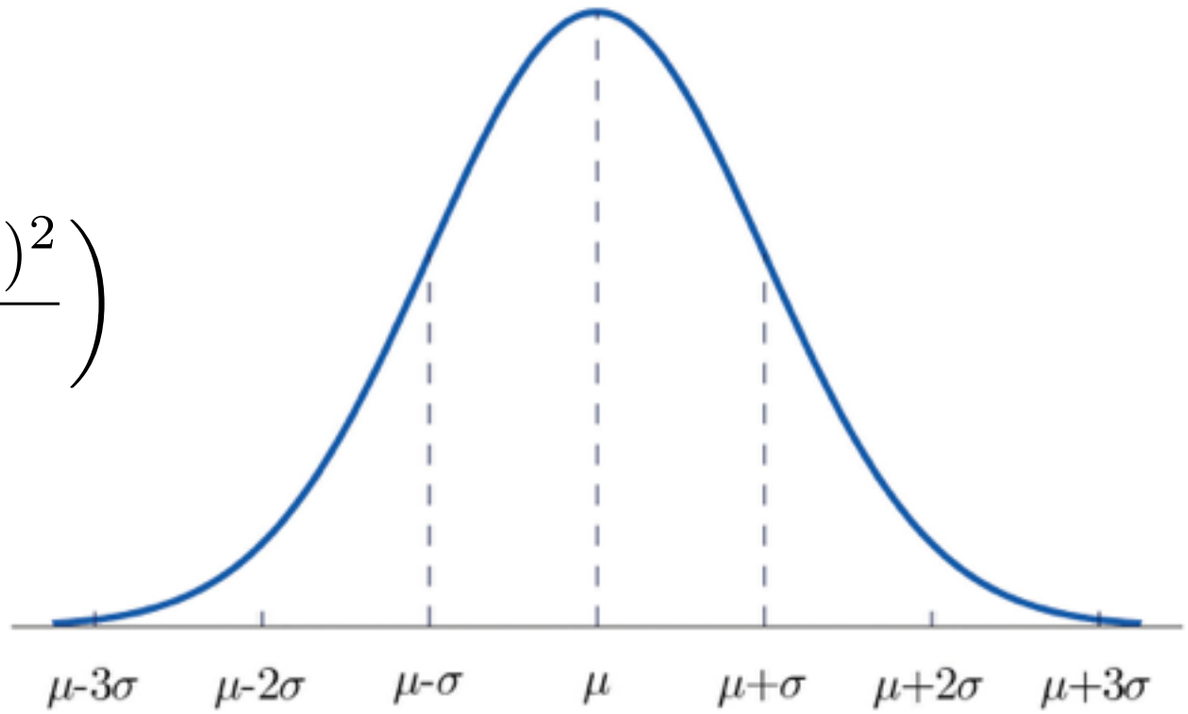
Gaussian model

- Let's try a more complex model involving correlation:

  Assume each image $x \in \mathbb{R}^{784}$ comes from a multivariate (784-dimensional) Gaussian distribution

- What do multivariate Gaussian distributions look like?
- Recall the univariate Gaussian $N(\mu, \sigma^2)$ has mean $\mu \in \mathbb{R}$ and variance $\sigma^2 > 0$

$$p(x) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

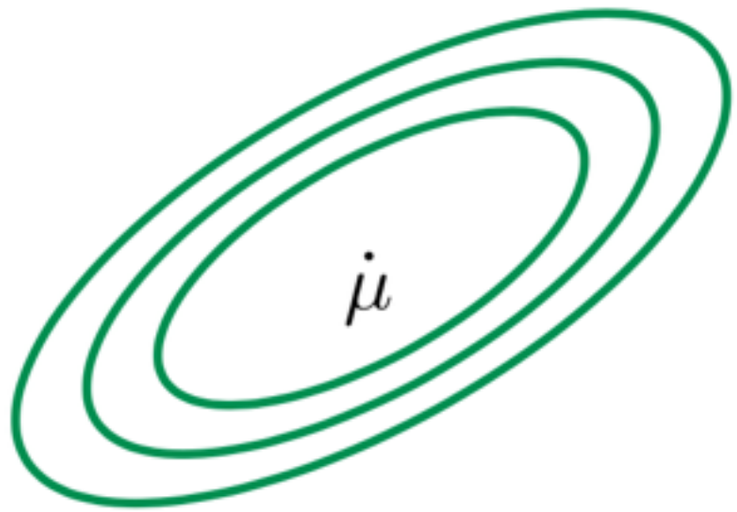


Multivariate Gaussian

- The Gaussian distribution $N(\mu, \Sigma)$ in $\mathbb{R}^d$ is specified by mean vector $\mu \in \mathbb{R}^d$ and $d \times d$ covariance matrix Σ

$$p(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$$

$(\text{where } |\Sigma| = \det(\Sigma))$



- Let $X = (X_1, \ldots, X_d)$ be a random draw from $N(\mu, \Sigma)$

  Then $\mathrm{E}(X) = \mu$. That is, $\mathrm{E}(X_i) = \mu_i$ for all $1 \le i \le d$.

  And $\mathrm{E}(X-\mu)(X-\mu)^\top = \Sigma$. That is, for all $1 \le i, j \le d$,

$$\mathrm{Cov}(X_i, X_j) = \mathrm{E}(X_i - \mu_i)(X_j - \mu_j) = \Sigma_{ij}$$

  In particular, $\mathrm{Var}(X_i) = \mathrm{E}(X_i - \mu_i)^2 = \Sigma_{ii}$.

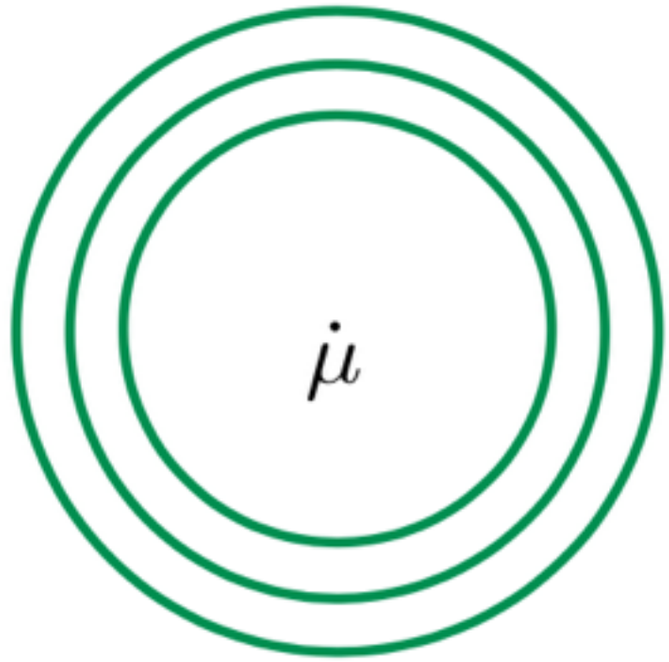
Special case: spherical Gaussian

$X_1, X_2, \ldots, X_d$ are independent with the same variance σ^2.
Thus, $\Sigma = \sigma^2 I_d$.

Simplified density:

$$p(x) = \frac{1}{(2\pi)^{d/2}\sigma^d} \exp\left(-\frac{\|x-\mu\|^2}{2\sigma^2}\right)$$

$$= \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i-\mu_i)^2}{2\sigma^2}\right)$$



Density at a point depends only on its distance from μ.

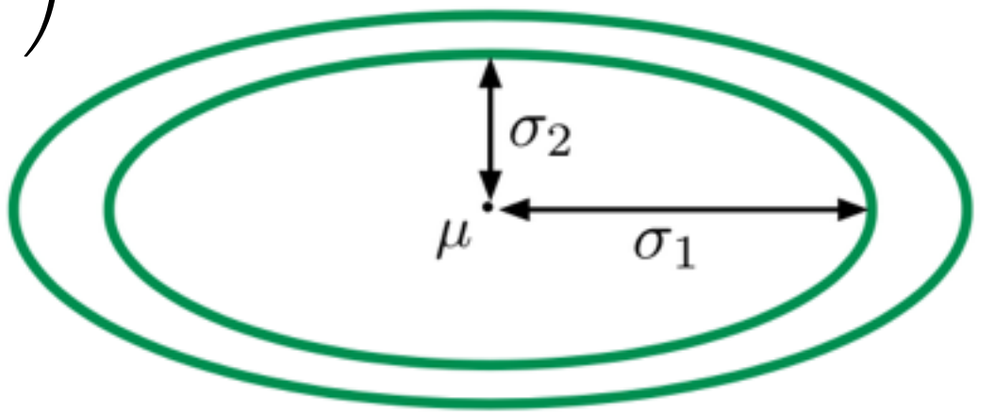
Each X_i is an independent univariate Gaussian $N(\mu_i, \sigma^2)$.

Special case: diagonal Gaussian

The X_i's are independent, with variances σ_i^2.
Thus, $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_d^2)$.

$$p(x) = \frac{1}{(2\pi)^{d/2}\sigma_1 \cdots \sigma_d} \exp\left(-\sum_{i=1}^{d} \frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right)$$

$$= \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right)$$



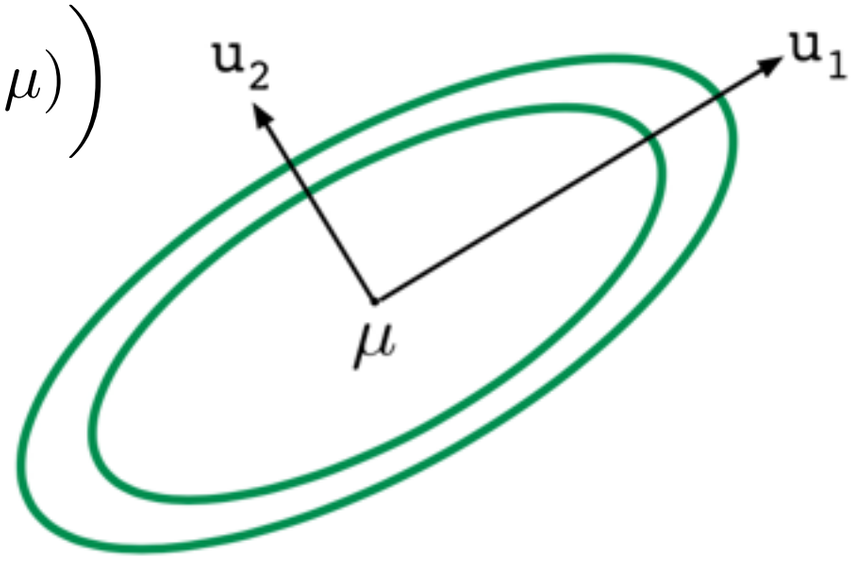
Contours of equal density are axis-aligned ellipsoids centered at μ.

Each X_i is an independent univariate Gaussian $N(\mu_i, \sigma_i^2)$.

The general Gaussian

$N(\mu, \Sigma)$ with general mean μ and covariance Σ:

$$p(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$$



Eigendecomposition of Σ:

- Eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d > 0$
- Corresponding eigenvectors $u_1, \ldots, u_d \in \mathbb{R}^d$

$N(\mu, \Sigma)$ is simply a rotated version of $N(\mu, \text{diag}(\lambda_1, \ldots, \lambda_d))$.

Fitting Gaussian to MNIST

- Assume in each class j, the conditional distribution $P_j(x)$ is Gaussian with mean $\mu_j \in \mathbb{R}^{784}$ and covariance matrix $\Sigma_j \in \mathbb{R}^{784 \times 784}$
- Estimate μ_j via the sample mean of the examples in class j:

$$\hat{\mu}_j = \frac{1}{n_j} \sum_{x \in \text{class}(j)} x$$

- Estimate Σ_j via the sample covariance of the examples in class j:

$$\hat{\Sigma}_j = \frac{1}{n_j} \sum_{x \in \text{class}(j)} (x - \hat{\mu}_j)(x - \hat{\mu}_j)^\top$$

- Formula for Gaussian density involves Σ_j^{-1}, may be singular
  - Need to regularize: $\hat{\Sigma}_j \to \hat{\Sigma}_j + \sigma^2 I$

Gaussian model classifier

- Data space $\mathcal{X} = \mathbb{R}^{784}$, label space $\mathcal{Y} = \{0, 1, \ldots, 9\}$
- From training data, estimate:

$$\text{prior } \pi_j, \ \text{mean } \mu_j, \ \text{covariance } \Sigma_j, \quad 0 \leq j \leq 9$$

- Given a (grayscale) image $x \in [0,1]^{784} \subseteq \mathbb{R}^{784}$, conditional log-likelihood is:

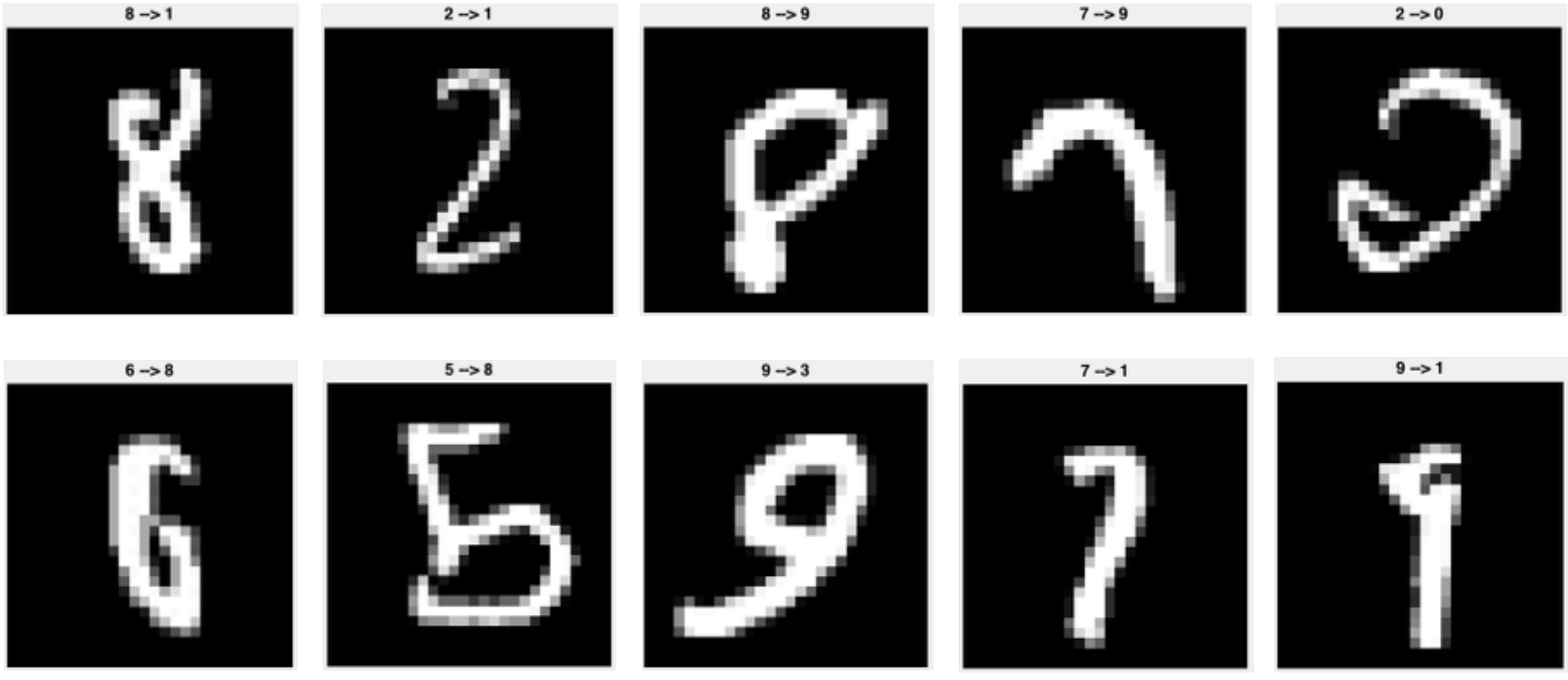
$$\log P_j(x) = -\frac{1}{2}(x - \mu_j)^\top \Sigma_j^{-1}(x - \mu_j) - \frac{1}{2}\log|\Sigma_j| - \frac{d}{2}\log(2\pi)$$

- Classify image x as:

$$\begin{aligned}\arg\max_j \ \pi_j \, P_j(x) &= \arg\max_j \ \log \pi_j + \log P_j(x) \\ &= \arg\max_j \ \log \pi_j - \frac{1}{2}(x - \mu_j)^\top \Sigma_j^{-1}(x - \mu_j) - \frac{1}{2}\log|\Sigma_j|\end{aligned}$$

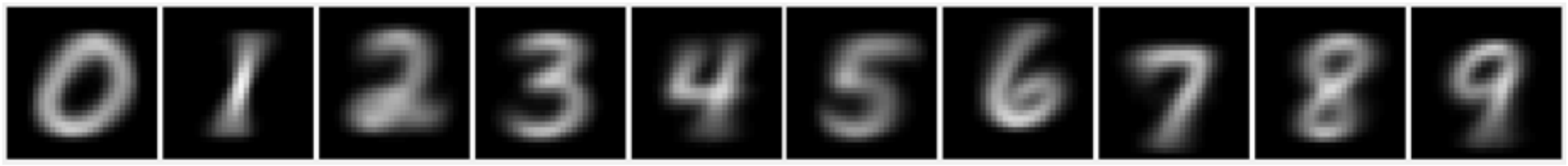
Gaussian model on MNIST

- Error rate: **4.58%** —> much better than naive Bayes (15.4%)!
- Some images that are misclassified:

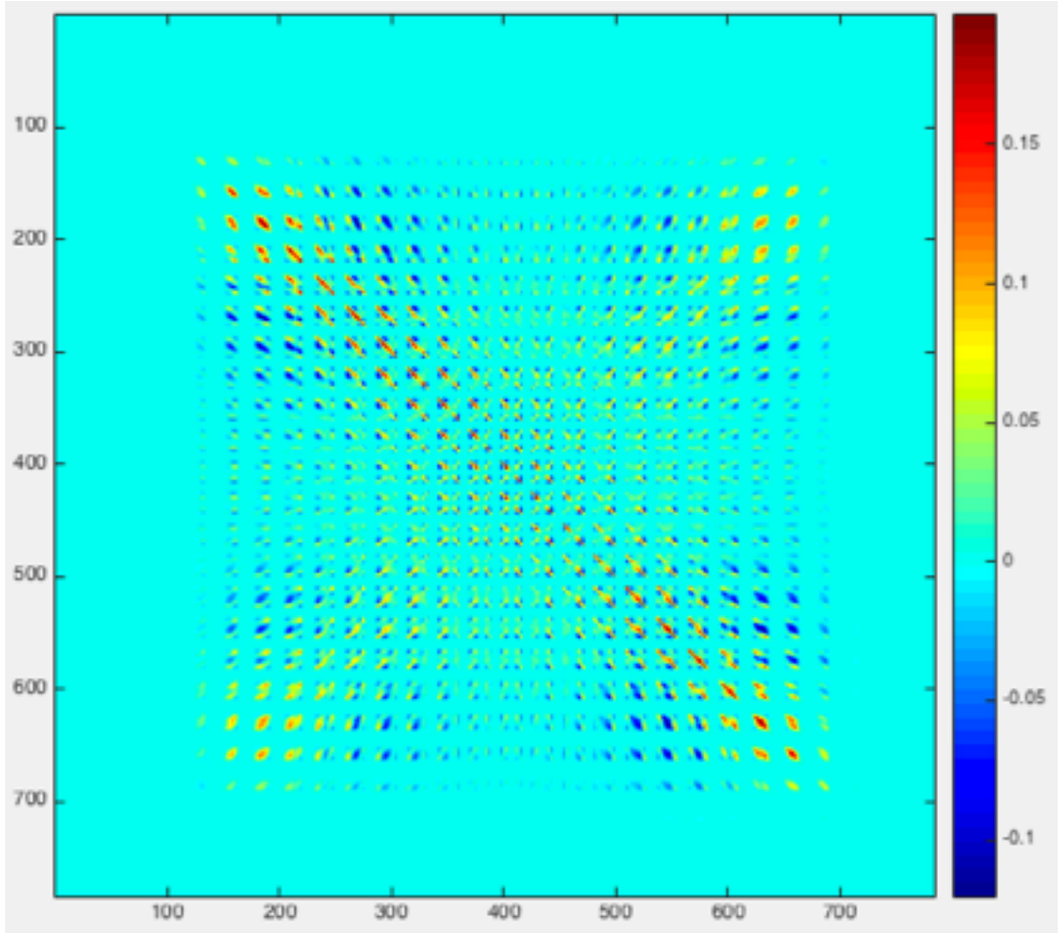


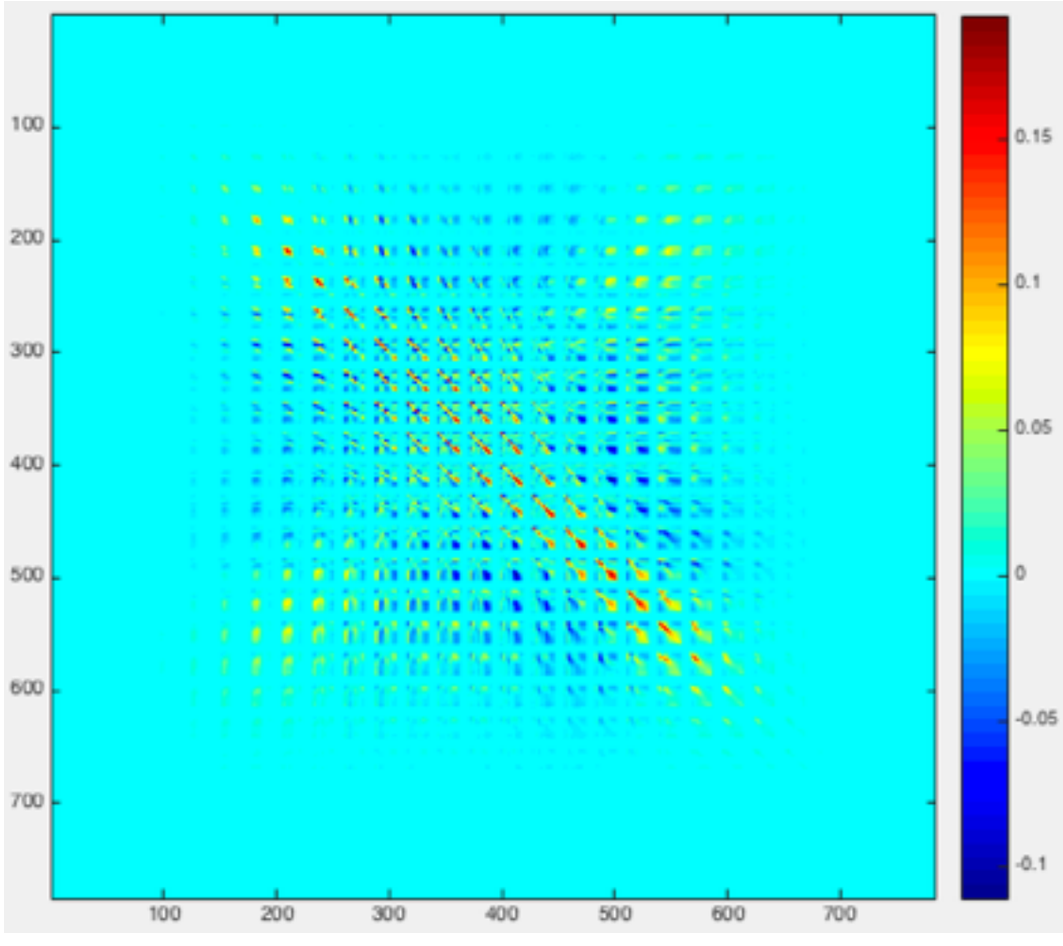
Gaussian model on MNIST

- Error rate: **4.58%** —> much better than naive Bayes (15.4%)!
- Mean vectors for each class (the μ_j's):



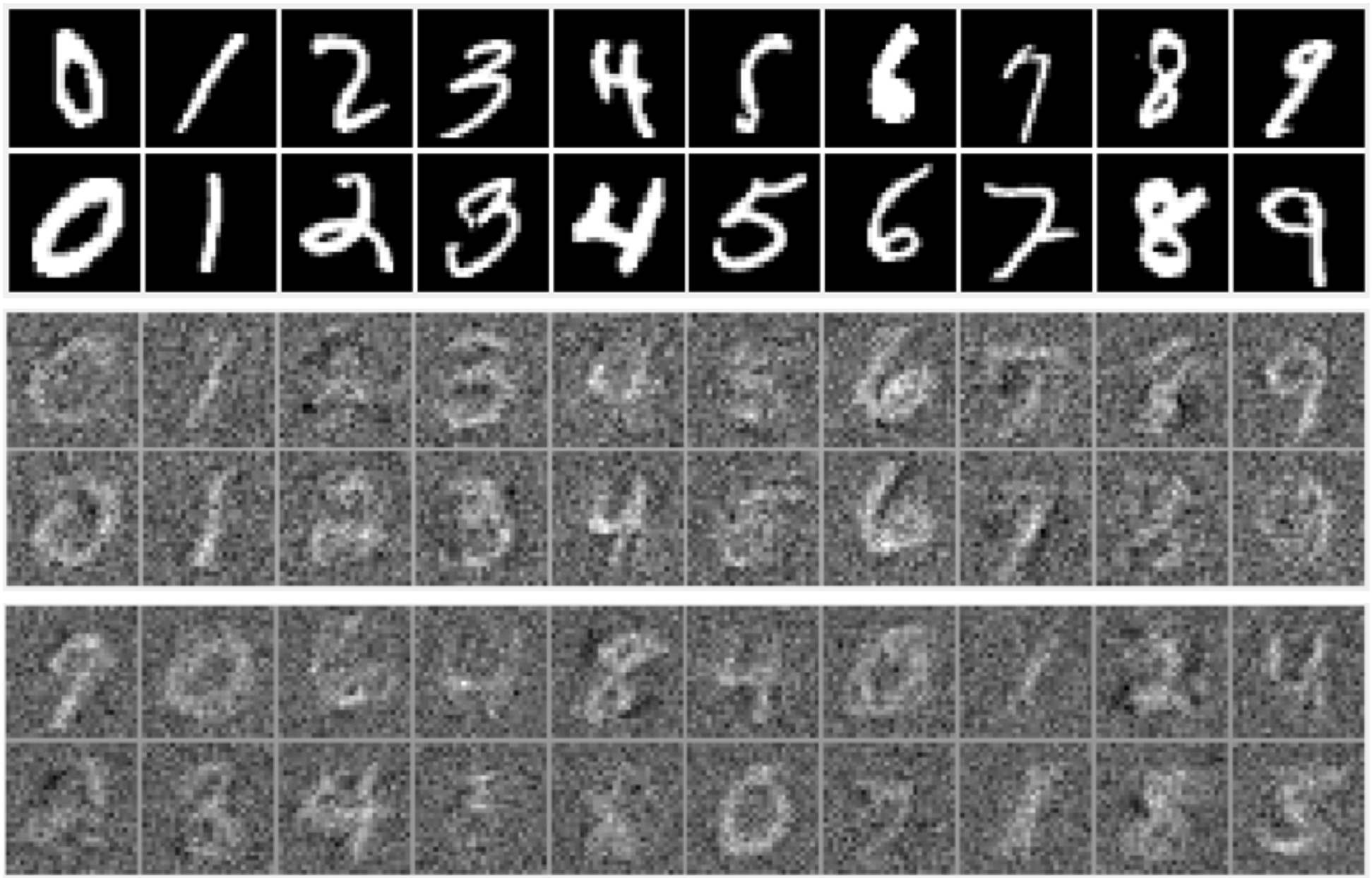
- Sample covariance matrix for class 0 (left) and class 9 (right):





Gaussian model on MNIST

- Compare data with samples generated from Gaussian model:

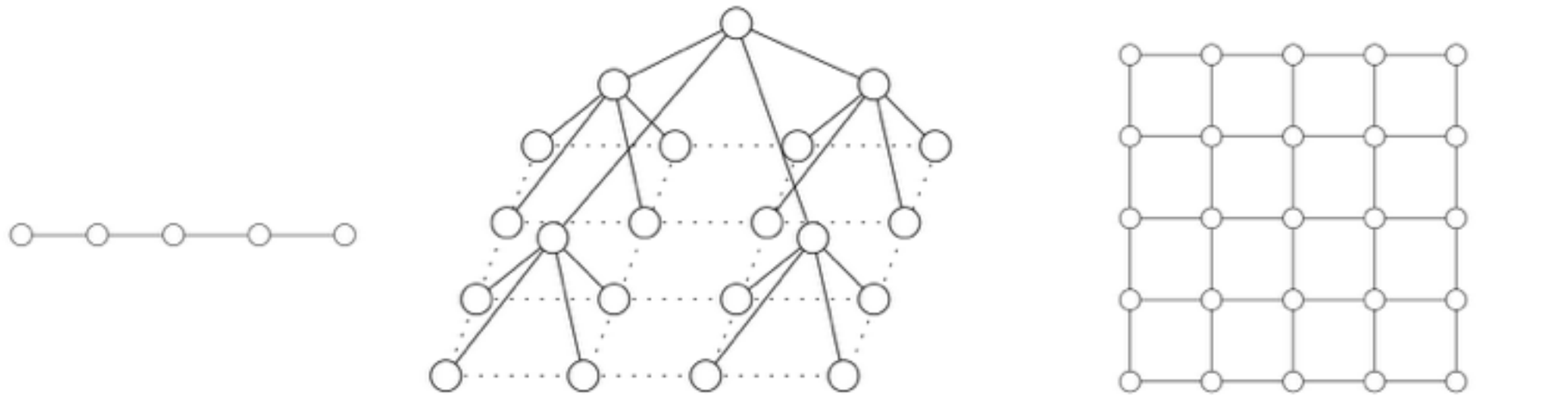


- Very noisy samples, and pixel values can even be negative!

Inference

- We have seen how to treat classification as inference
- Probabilistic view: generative model of the world
- Can make simple, unrealistic model, but in conjunction with Bayesian analysis, still performs pretty well
- Probabilistic graphical model: graph theory + probability

  (Naive Bayes = graph with no edges)



(a) Markov chain (b) Multiscale quadtree (c) Two-dimensional grid