香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

COMP 5212

Machine Learning

Lecture 8

Generative Models, Naive Bayes

Junxian He

Feb 28, 2024

Clarification on HWs / Exams

- Future homeworks will be less workloaded

  The bitter lesson is HW difficulty is always positively correlated with reward

- Exams will be easier than HWs (different formats)

  No point to make exams difficult

- No worry on GPAs as long as you are trying to learn and write the homework yourself

Discriminative vs. Generative Learning

X

$p(y \mid x)$ Discriminative

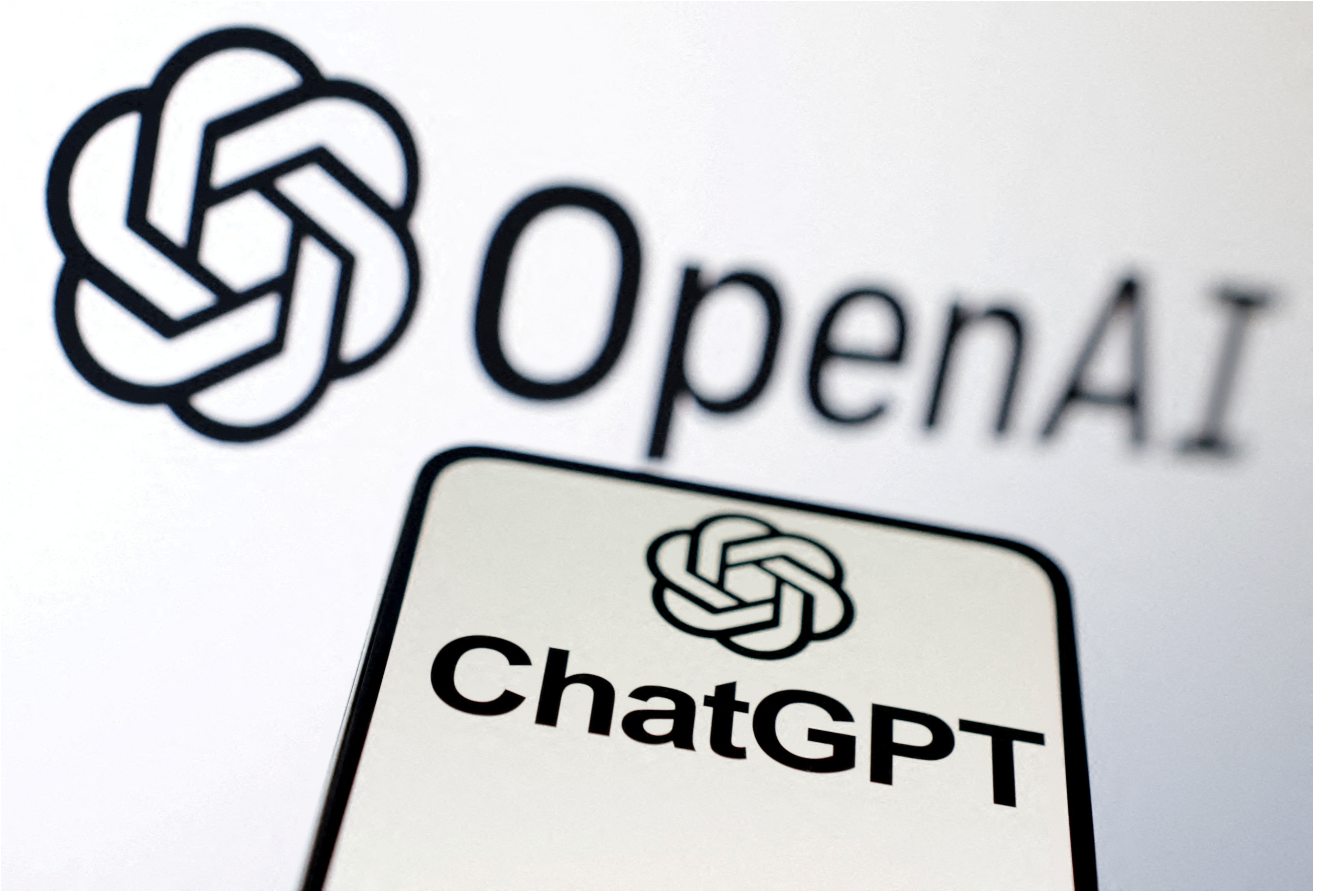
Cat Y

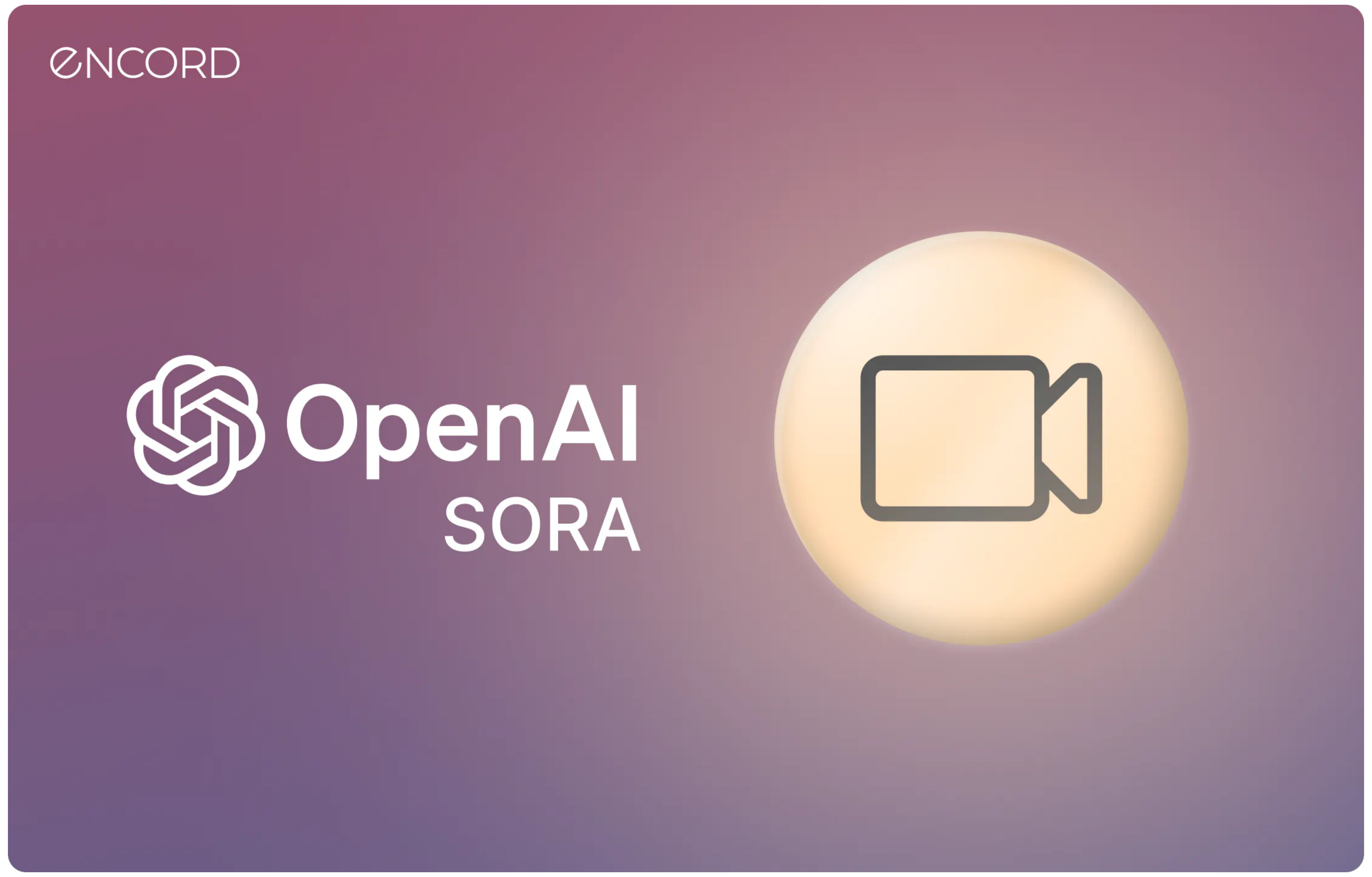
Cat Y $p(y)$

Generative $p(x \mid y)$

X

Generative Model Examples





Video Generation Examples

Prompt: A stylish woman walks down a Tokyo street filled with warm glowing neon and animated city signage. She wears a black leather jacket, a long red dress, and black boots, and carries a black purse. She wears sunglasses and red lipstick. She walks confidently and casually. The street is damp and reflective, creating a mirror effect of the colorful lights. Many pedestrians walk about.

Video Generation Examples

Prompt: Photorealistic closeup video of two pirate ships battling each other as they sail inside a cup of coffee.

Video Generation Examples

Prompt: A petri dish with a bamboo forest growing within it that has tiny red pandas running around.

Discriminative vs. Generative Learning

X

$p(y \mid x)$

Discriminative

Cat Y

Cat Y $p(y)$

Generative $p(x \mid y)$

X

Bayes Rule

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

$$p(x) = \sum_y p(x,y) = \sum_y p(x\,|\,y)p(y)$$

If our goal is to predict y, the distribution is often written as:

$$p(y\,|\,x) \propto p(x\,|\,y)p(y)$$

$$\begin{aligned} \arg\max_y p(y|x) &= \arg\max_y \frac{p(x|y)p(y)}{p(x)} \\ &= \arg\max_y p(x|y)p(y). \end{aligned}$$

Generative Models Compared to Discriminative Models

Pros:

- Generative models can generate data (generation, data augmentation)
- Inject prior information through the prior distribution
- May be learned in an unsupervised way when y is not available
- Modeling data distribution is a fundamental goal in AI

Cons:

- Often underperforms discriminative models on discriminative tasks because of stronger assumptions on the data

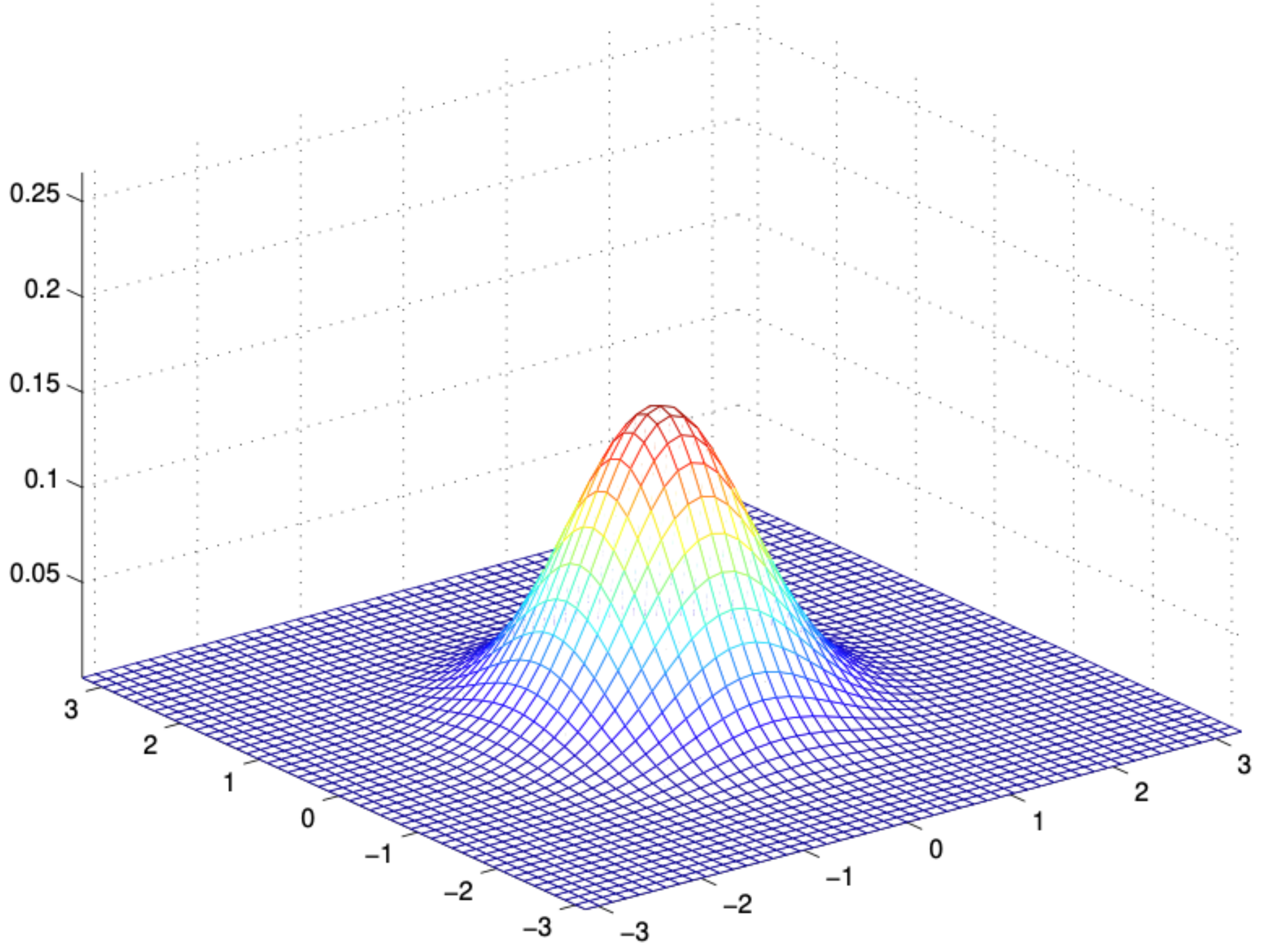
Gaussian Discriminant Analysis Model (GDA)

Multivariate Gaussian distribution

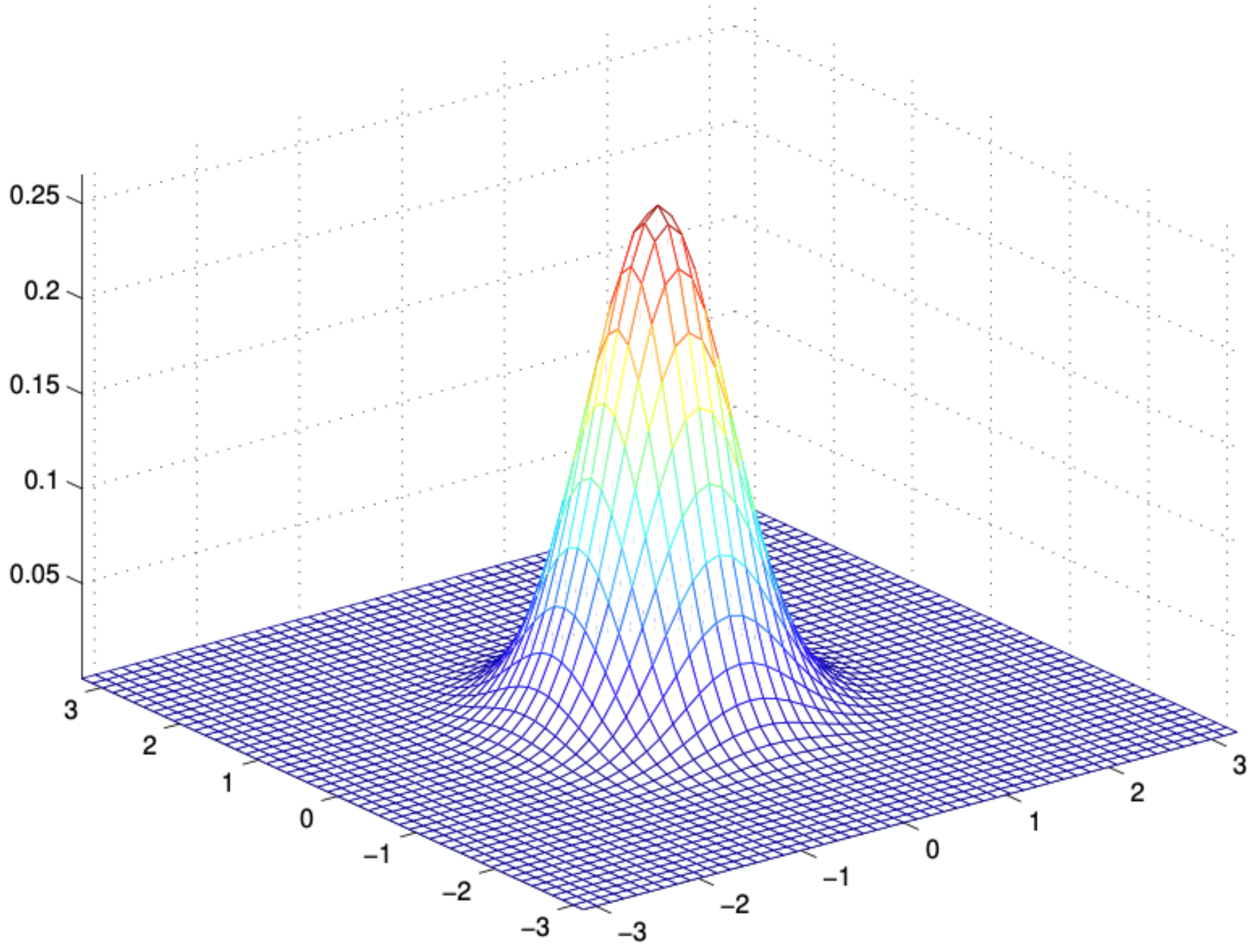
$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

- $\Sigma \in R^{dxd}$ is the covariance matrix, it is also symmetric positive semi-definite
- $|\Sigma|$ denotes the determinant of Σ

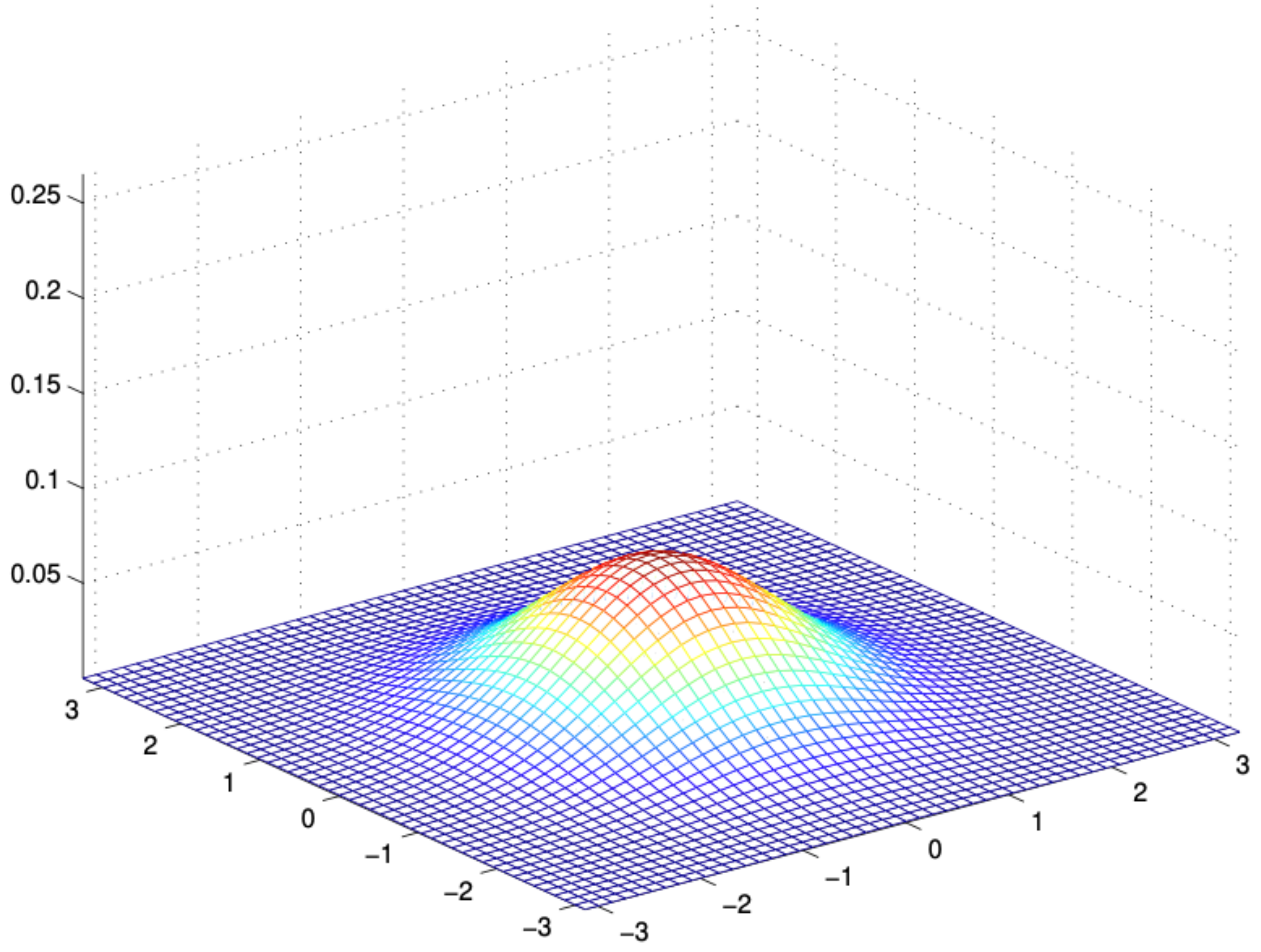
Examples of Multivariate Gaussian



$\Sigma = I$

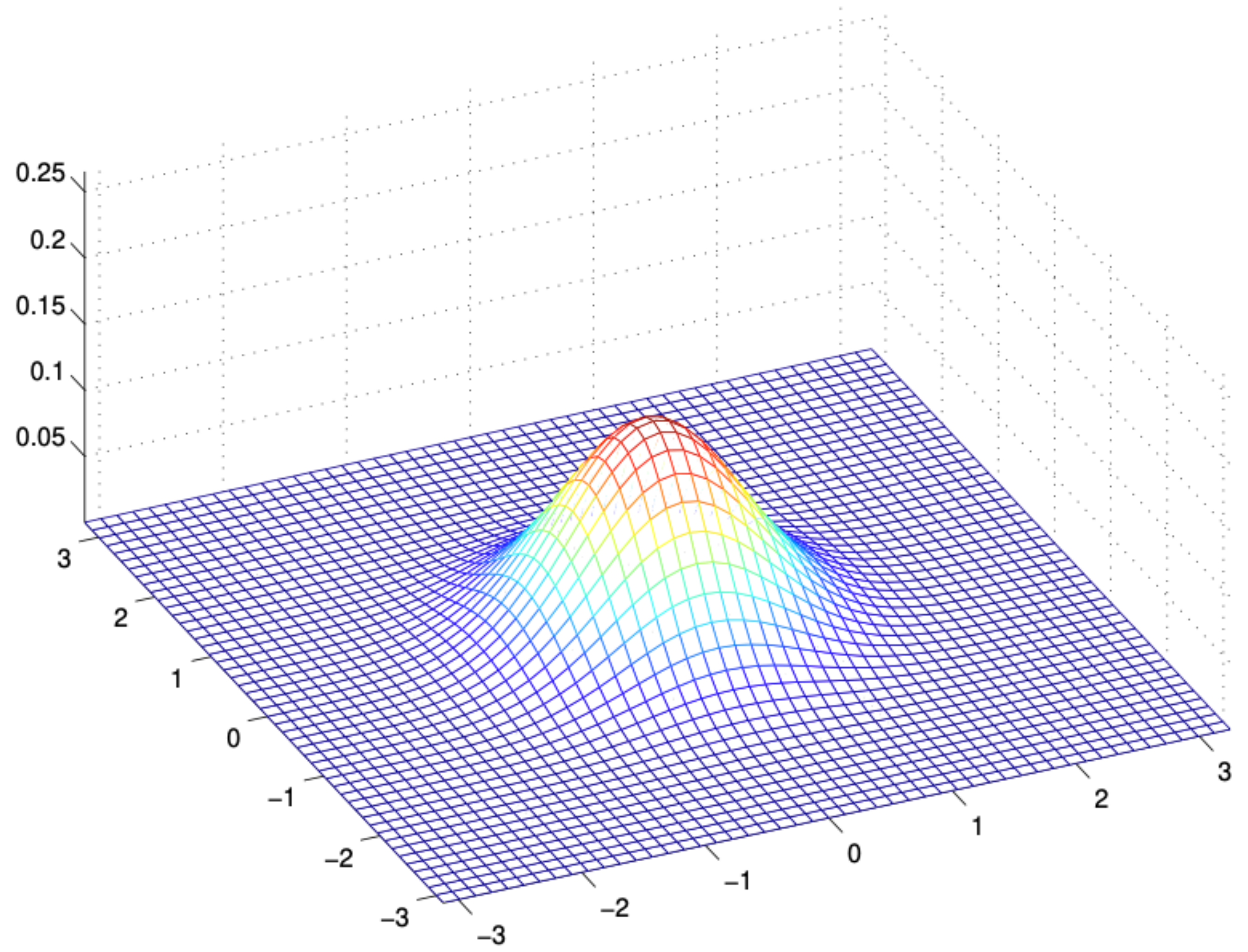


$\Sigma = 0.6I$

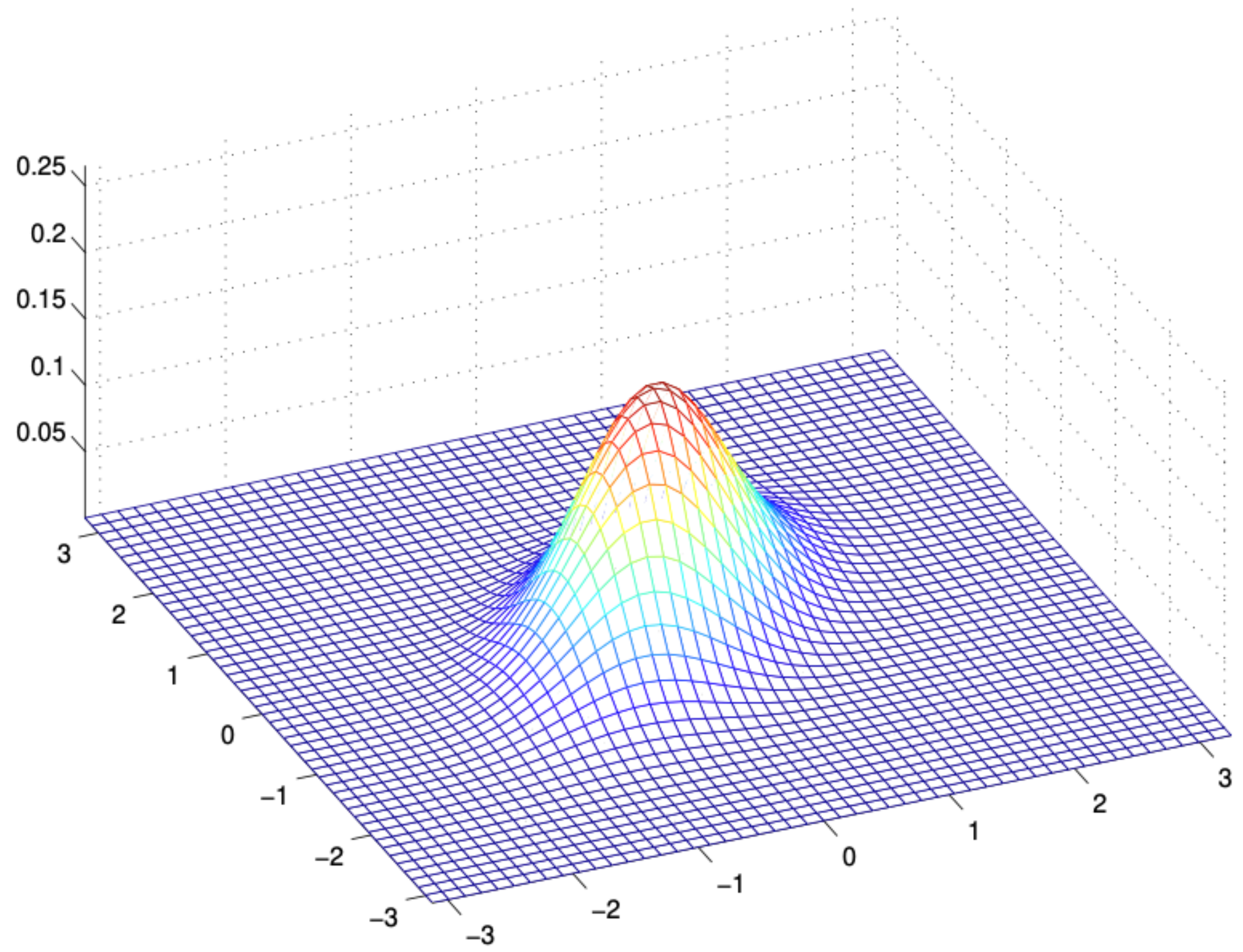


$\Sigma = 2I$

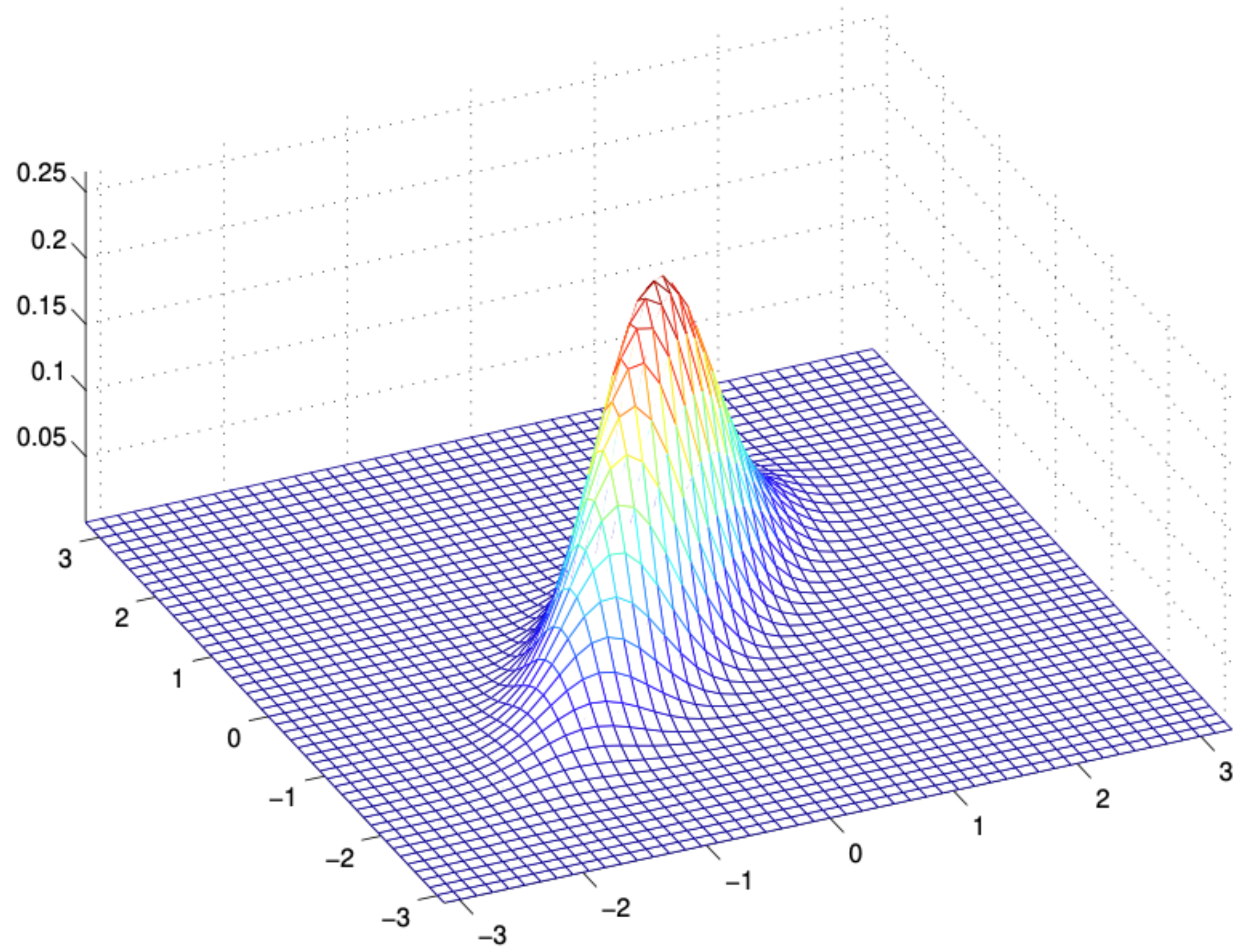
Examples of Multivariate Gaussian



$$\Sigma = \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right]$$



$$\Sigma = \left[\begin{array}{cc} 1 & 0.5 \\ 0.5 & 1 \end{array} \right]$$



$$\Sigma = \left[\begin{array}{cc} 1 & 0.8 \\ 0.8 & 1 \end{array} \right]$$

Gaussian Discriminant Analysis Model

Binary classification: $y \in \{0,1\}, x \in R^d$

Assumption

$$
\begin{aligned}
y &\sim \text{Bernoulli}(\phi) \\
x|y=0 &\sim \mathcal{N}(\mu_0, \Sigma) \\
x|y=1 &\sim \mathcal{N}(\mu_1, \Sigma)
\end{aligned}
$$

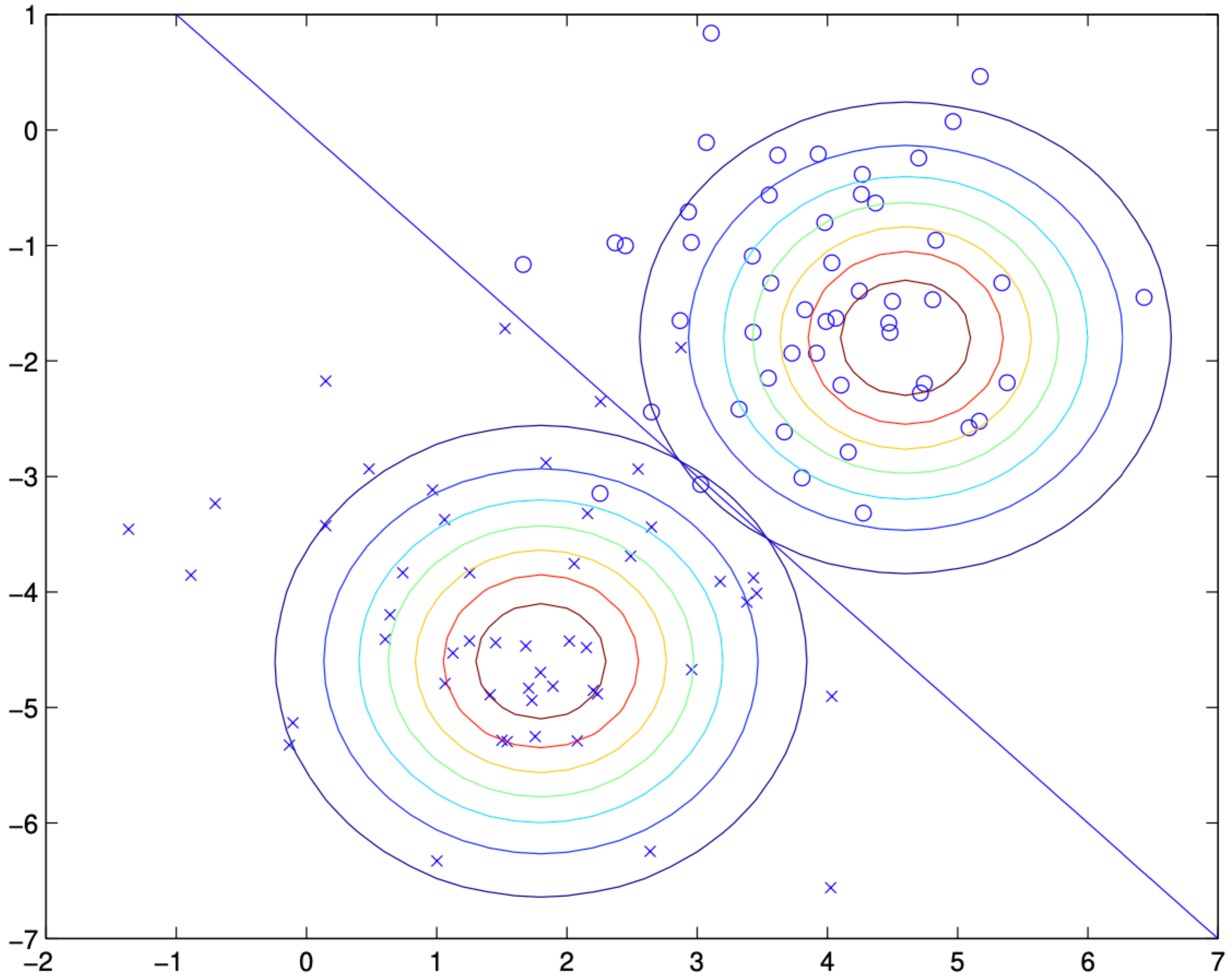
$$
\begin{aligned}
p(y) &= \phi^y(1-\phi)^{1-y} \\
p(x|y=0) &= \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\right) \\
p(x|y=1) &= \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)
\end{aligned}
$$

Maximum Likelihood Estimation

$$\begin{aligned}\ell(\phi, \mu_0, \mu_1, \Sigma) &= \log \prod_{i=1}^{n} p(x^{(i)}, y^{(i)}; \phi, \mu_0, \mu_1, \Sigma) \\ &= \log \prod_{i=1}^{n} p(x^{(i)}|y^{(i)}; \mu_0, \mu_1, \Sigma) p(y^{(i)}; \phi).\end{aligned}$$

Why is the decision boundary linear?

$$\begin{aligned}\phi &= \frac{1}{n} \sum_{i=1}^{n} 1\{y^{(i)} = 1\} \\ \mu_0 &= \frac{\sum_{i=1}^{n} 1\{y^{(i)} = 0\} x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = 0\}} \\ \mu_1 &= \frac{\sum_{i=1}^{n} 1\{y^{(i)} = 1\} x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = 1\}} \\ \Sigma &= \frac{1}{n} \sum_{i=1}^{n} (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T\end{aligned}$$



Connection Between GDA and Logistic Regression

Through Bayes rule, we can show that

$$p(y=1|x;\phi,\Sigma,\mu_0,\mu_1)=\frac{1}{1+\exp(-\theta^T x)}$$

$$\theta = f(\phi, \Sigma, \mu_0, \mu_1)$$

$p(x\,|\,y)$ is Gaussian $\Rightarrow$ $p(y\,|\,x)$ follows logistic regression

$p(x\,|\,y)$ is Gaussian $\not\Leftarrow$ $p(y\,|\,x)$ follows logistic regression

Gaussian Discriminative Analysis model makes stronger assumptions

Connection Between GDA and Logistic Regression

Gaussian Discriminative Analysis (GDA) model makes stronger assumptions

- When x|y does not follow Gaussian in practice, GDA may or may not do well
- When x|y does not follow Gaussian and the training data is large, the method that makes weaker assumptions (logistic regression) will always do better
- When x|y indeed follows Gaussian and the training data is small, the method that makes stronger assumptions will do well (more data-efficient)

These are intuitions generally applicable to machine learning

Philosophy Behind Modeling Assumptions / Priors

- When x|y does not follow Gaussian in practice, GDA may or may not do well
- When x|y does not follow Gaussian and the training data is large, the method that makes weaker assumptions (logistic regression) will always do better
- When x|y indeed follows Gaussian and the training data is small, the method that makes stronger assumptions will do well (more data-efficient)

1. Transformers v.s. LSTMs v.s. CNN. — transformers can be worse on small data, but stand out with large data (pretraining)

2. The famous and bitter lesson from IBM machine translation model: "Every time I fire a linguist, the model performance goes up" — Frederick Jelinek

Naive Bayes

Binary classification: $y \in \{0,1\}$, x is discrete

Consider an email spam detection task, to predict whether the email is spam or not

How to represent the text?

if an email contains the j-th word of the dictionary, then we will set $x_j = 1$; otherwise, we let $x_j = 0$

$$x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \quad \begin{array}{l} \text{a} \\ \text{aardvark} \\ \text{aardwolf} \\ \vdots \\ \text{buy} \\ \vdots \\ \text{zygmurgy} \end{array}$$

vocabulary

Dimension is the size of the dictionary

Email Spam Classification

$$x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \text{a} \\ \text{aardvark} \\ \text{aardwolf} \\ \vdots \\ \text{buy} \\ \vdots \\ \text{zygmurgy} \end{array}$$

Suppose the dictionary has 50000 words, how many possible x?

Naive Bayes assumption: x_i's are conditionally independent given y

For any i and j, $p(x_i \mid y) = p(x_i \mid y, x_j)$

Email Spam Classification

Autoregressive

$$
\begin{aligned}
& p(x_1, \ldots, x_{50000}|y) \\
= \;& \boxed{p(x_1|y)p(x_2|y, x_1)p(x_3|y, x_1, x_2) \cdots p(x_{50000}|y, x_1, \ldots, x_{49999})} \\
= \;& p(x_1|y)p(x_2|y)p(x_3|y) \cdots p(x_{50000}|y) \\
= \;& \prod_{j=1}^{d} p(x_j|y)
\end{aligned}
$$

Parameters

$$\phi_{j|y=1} = p(x_j = 1\,|\,y = 1), \;\; \phi_{j|y=1} = p(x_j = 1\,|\,y = 0), \;\; \phi_y = p(y = 1)$$

50000 x 2 + 1 parameters (dict size is 50000)

Maximum Likelihood Estimation

$$\mathcal{L}(\phi_y, \phi_{j|y=0}, \phi_{j|y=1}) = \prod_{i=1}^{n} p(x^{(i)}, y^{(i)})$$

$$\begin{aligned}
\phi_{j|y=1} &= \frac{\sum_{i=1}^{n} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\}}{\sum_{i=1}^{n} 1\{y^{(i)} = 1\}} \\
\phi_{j|y=0} &= \frac{\sum_{i=1}^{n} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\}}{\sum_{i=1}^{n} 1\{y^{(i)} = 0\}} \\
\phi_y &= \frac{\sum_{i=1}^{n} 1\{y^{(i)} = 1\}}{n}
\end{aligned}$$

Count the occurrence of x_j in spam/ non-spam emails and normalize

Prediction

$$
\begin{aligned}
p(y=1|x) &= \frac{p(x|y=1)p(y=1)}{p(x)} \\
&= \frac{\left(\prod_{j=1}^{d} p(x_j|y=1)\right) p(y=1)}{\left(\prod_{j=1}^{d} p(x_j|y=1)\right) p(y=1) + \left(\prod_{j=1}^{d} p(x_j|y=0)\right) p(y=0)}
\end{aligned}
$$

Naive Classifier

Laplace Smoothing

What if we never see the word "learning" in training data but "learning" exists in the test data?

$$\phi_{j|y=1} = \frac{\sum_{i=1}^{n} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\}}{\sum_{i=1}^{n} 1\{y^{(i)} = 1\}}$$

$$\phi_{j|y=0} = \frac{\sum_{i=1}^{n} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\}}{\sum_{i=1}^{n} 1\{y^{(i)} = 0\}}$$

Suppose the index in the dictionary for "learning" is q

$$p(x_q = 1 \,|\, y = 1) = 0$$

$$p(x_q = 1 \,|\, y = 0) = 0$$

$$\begin{aligned} p(y=1|x) &= \frac{p(x|y=1)p(y=1)}{p(x)} \\ &= \frac{\left(\prod_{j=1}^{d} p(x_j|y=1)\right) p(y=1)}{\left(\prod_{j=1}^{d} p(x_j|y=1)\right) p(y=1) + \left(\prod_{j=1}^{d} p(x_j|y=0)\right) p(y=0)} = \frac{0}{0} \end{aligned}$$

Laplace Smoothing

Take the problem of estimating the mean of a multinomial random variable z taking values in {1, …, k}. Given the independent observations $\{z^{(1)}, \cdots, z^{(n)}\}$

$$\phi_j = p(z = j) \qquad \phi_j = \frac{\sum_{i=1}^{n} 1\{z^{(i)} = j\}}{n}$$

$$\phi_j = \frac{1 + \sum_{i=1}^{n} 1\{z^{(i)} = j\}}{k + n}$$

Why adding k to the denominator?

In the email spam classification case:

$$\phi_{j|y=1} = \frac{1 + \sum_{i=1}^{n} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\}}{2 + \sum_{i=1}^{n} 1\{y^{(i)} = 1\}}$$

$$\phi_{j|y=0} = \frac{1 + \sum_{i=1}^{n} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\}}{2 + \sum_{i=1}^{n} 1\{y^{(i)} = 0\}}$$

Thank You!
Q & A