



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
Oct 3, 2024

Recap: The Dual Problem of SVM

Recap: The Dual Problem of SVM

$$\begin{aligned} \max_{\alpha} \quad & W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle \\ \text{s.t.} \quad & \alpha_i \geq 0, \quad i = 1, \dots, n \\ & \sum_{i=1}^n \alpha_i y^{(i)} = 0, \end{aligned}$$

Recap: The Dual Problem of SVM

$$\max_{\alpha} \quad W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle$$

$$\text{s.t. } \alpha_i \geq 0, \quad i = 1, \dots, n$$

$$\sum_{i=1}^n \alpha_i y^{(i)} = 0,$$

Kernel is all we need!

Recap: The Dual Problem of SVM

$$\max_{\alpha} \quad W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle$$

$$\text{s.t. } \alpha_i \geq 0, \quad i = 1, \dots, n$$

$$\sum_{i=1}^n \alpha_i y^{(i)} = 0,$$

Kernel is all we need!

After solving α (coordinate ascent with clipping, 6.8.2 of the CS229 Notes)

Recap: The Dual Problem of SVM

$$\max_{\alpha} \quad W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle$$

$$\text{s.t. } \alpha_i \geq 0, \quad i = 1, \dots, n$$

$$\sum_{i=1}^n \alpha_i y^{(i)} = 0,$$

Kernel is all we need!

After solving α (coordinate ascent with clipping, 6.8.2 of the CS229 Notes)

$$w = \sum_{i=1}^n \alpha_i y^{(i)} x^{(i)}$$

Recap: The Dual Problem of SVM

$$\max_{\alpha} \quad W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle$$

$$\text{s.t. } \alpha_i \geq 0, \quad i = 1, \dots, n$$

$$\sum_{i=1}^n \alpha_i y^{(i)} = 0,$$

Kernel is all we need!

After solving α (coordinate ascent with clipping, 6.8.2 of the CS229 Notes)

$$w = \sum_{i=1}^n \alpha_i y^{(i)} x^{(i)}$$

From KKT Conditions

Recap: The Dual Problem of SVM

$$\max_{\alpha} \quad W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle$$

$$\text{s.t. } \alpha_i \geq 0, \quad i = 1, \dots, n$$

$$\sum_{i=1}^n \alpha_i y^{(i)} = 0,$$

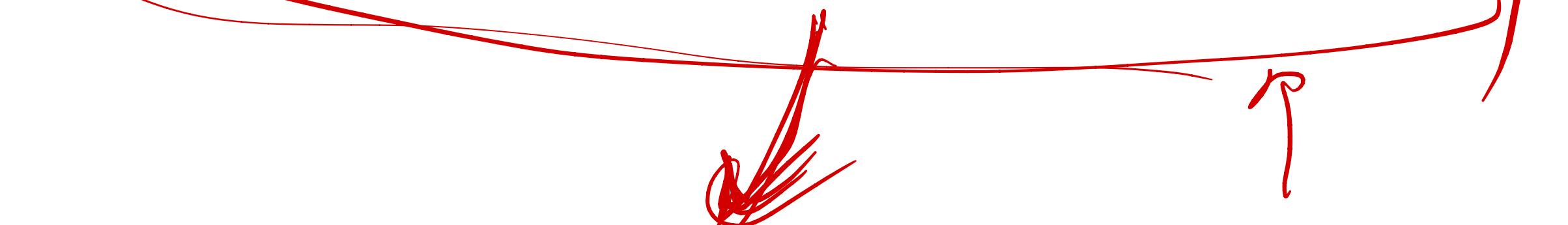
Kernel is all we need!

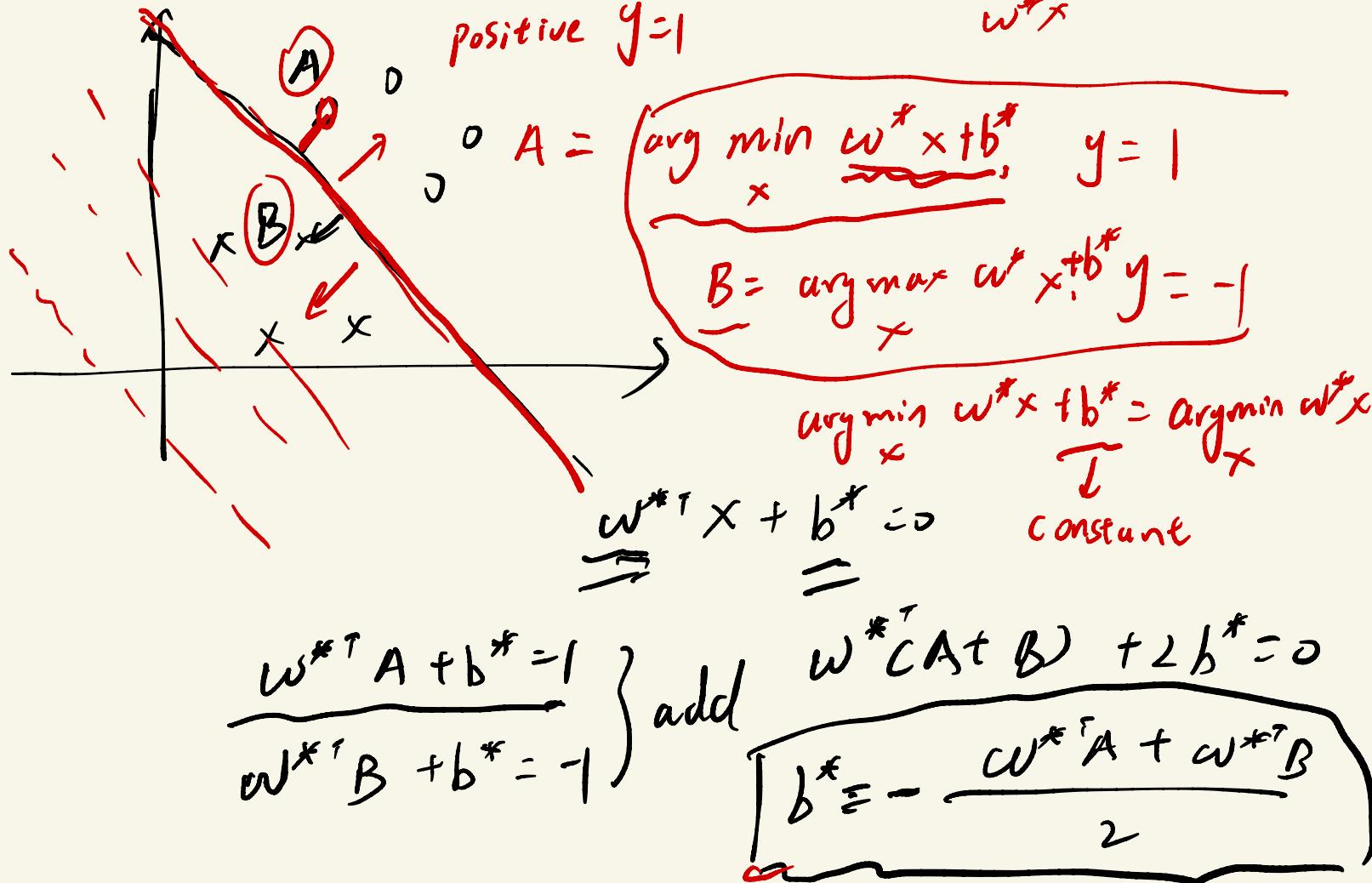
After solving α (coordinate ascent with clipping, 6.8.2 of the CS229 Notes)

$$w = \sum_{i=1}^n \alpha_i y^{(i)} x^{(i)}$$

From KKT Conditions

$$b^* = -\frac{\max_{i:y^{(i)}=-1} w^{*T} x^{(i)} + \min_{i:y^{(i)}=1} w^{*T} x^{(i)}}{2}$$





Recap: The Dual Problem of SVM

$$\max_{\alpha} \quad W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle$$

$$\text{s.t. } \alpha_i \geq 0, \quad i = 1, \dots, n$$

$$\sum_{i=1}^n \alpha_i y^{(i)} = 0,$$

Kernel is all we need!

After solving α (coordinate ascent with clipping, 6.8.2 of the CS229 Notes)

$$w = \sum_{i=1}^n \alpha_i y^{(i)} x^{(i)}$$

From KKT Conditions

$$b^* = -\frac{\max_{i:y^{(i)}=-1} w^{*T} x^{(i)} + \min_{i:y^{(i)}=1} w^{*T} x^{(i)}}{2}$$

From the original constraints

Discriminative vs. Generative Learning

Discriminative vs. Generative Learning



X

$$p(y | x)$$

Discriminative

Cat Y

Discriminative vs. Generative Learning



$$p(y | x)$$

Discriminative

Cat

Y

$$\underline{P(y | x)}$$

X

Generative

Cat

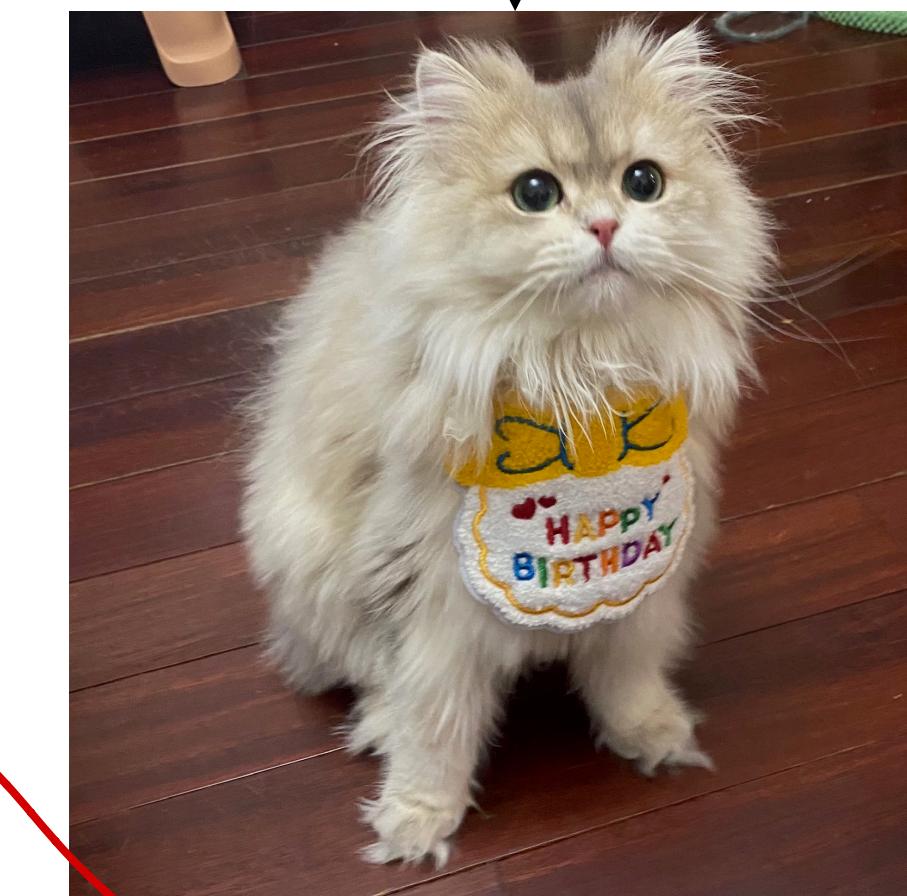
$$Y \ p(y)$$

$$P(y)$$

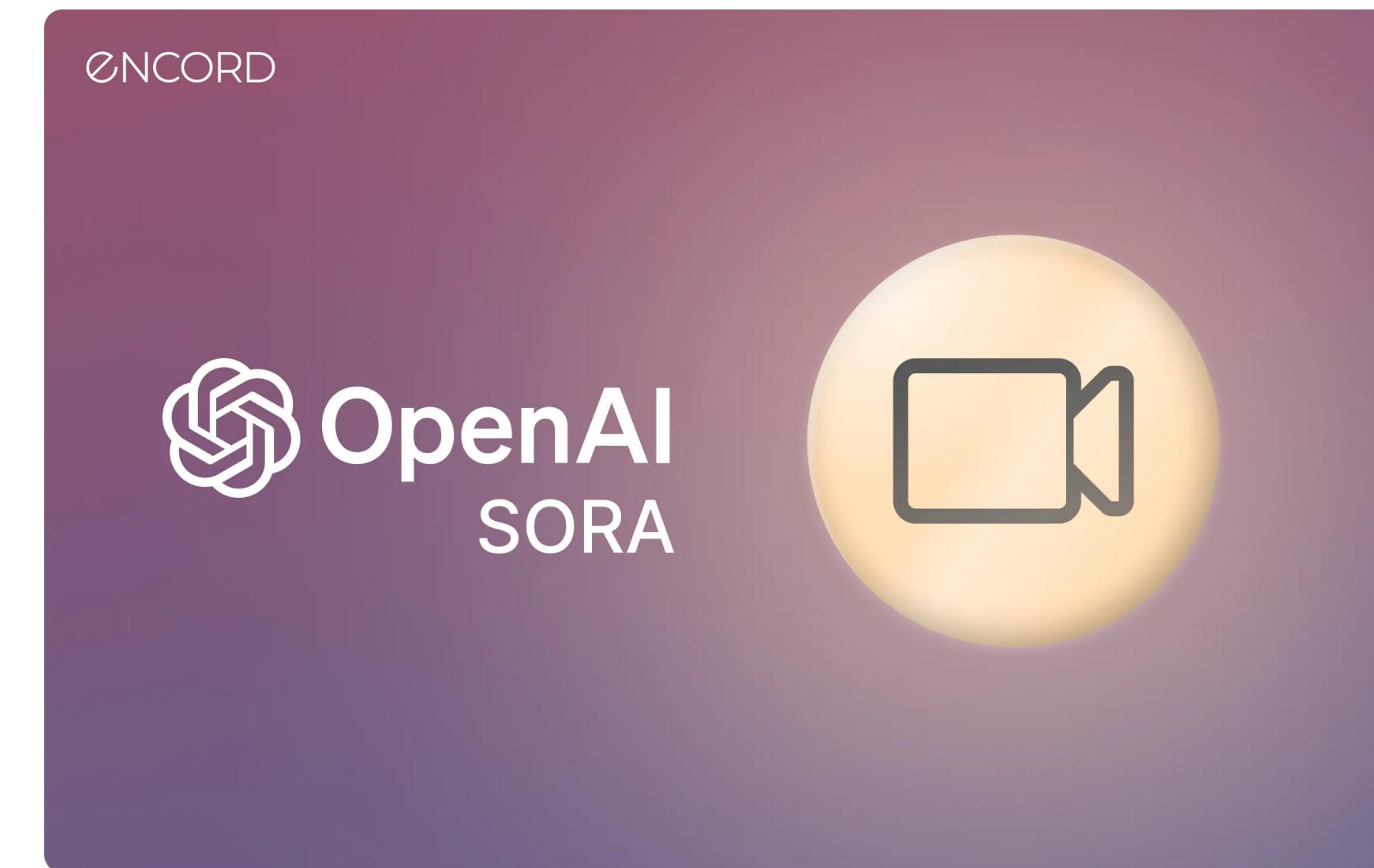
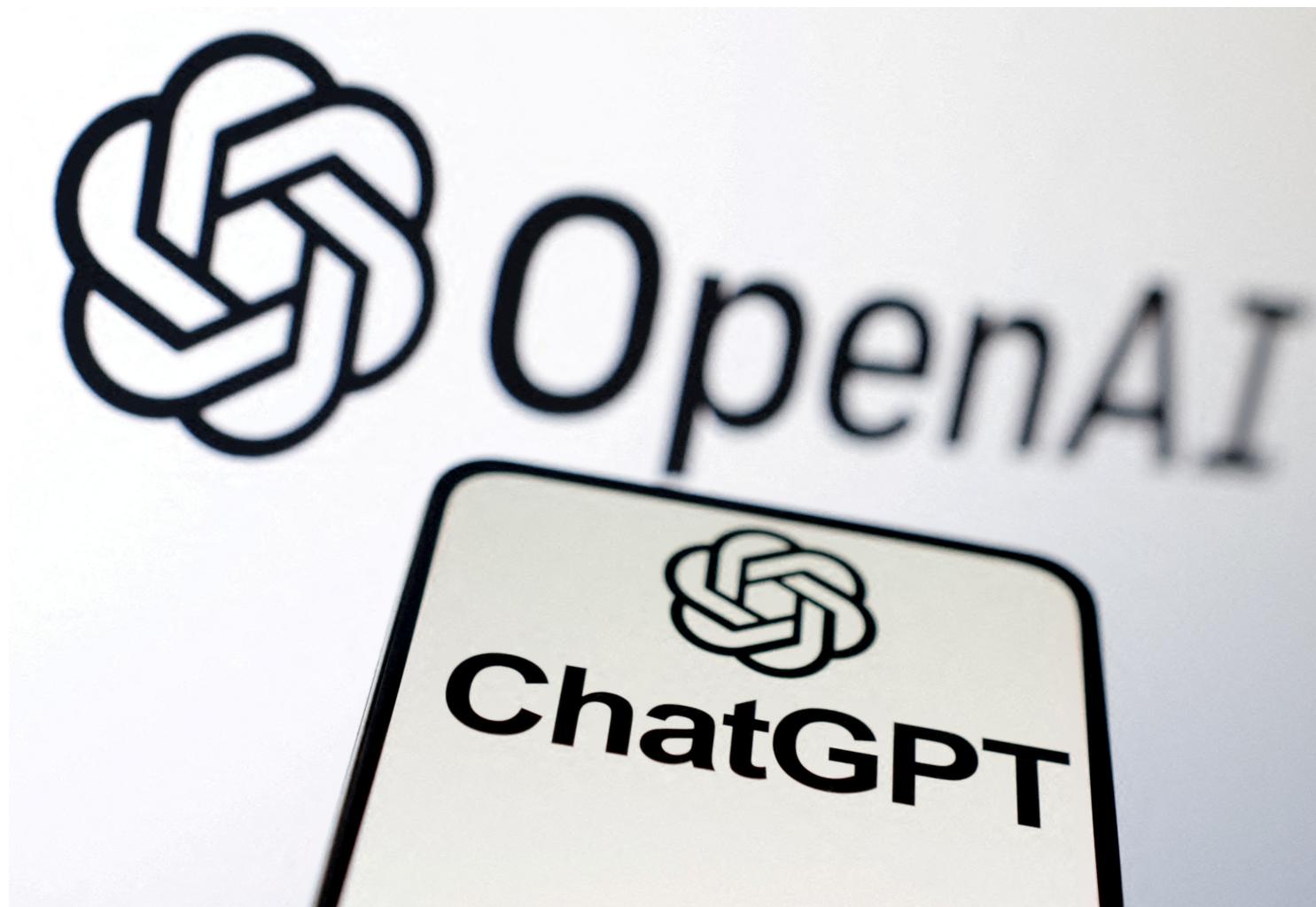
$$p(x | y)$$

$$P_{x|y}$$

X



Generative Model Examples



$P(x)$ \times language

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: 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: 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.

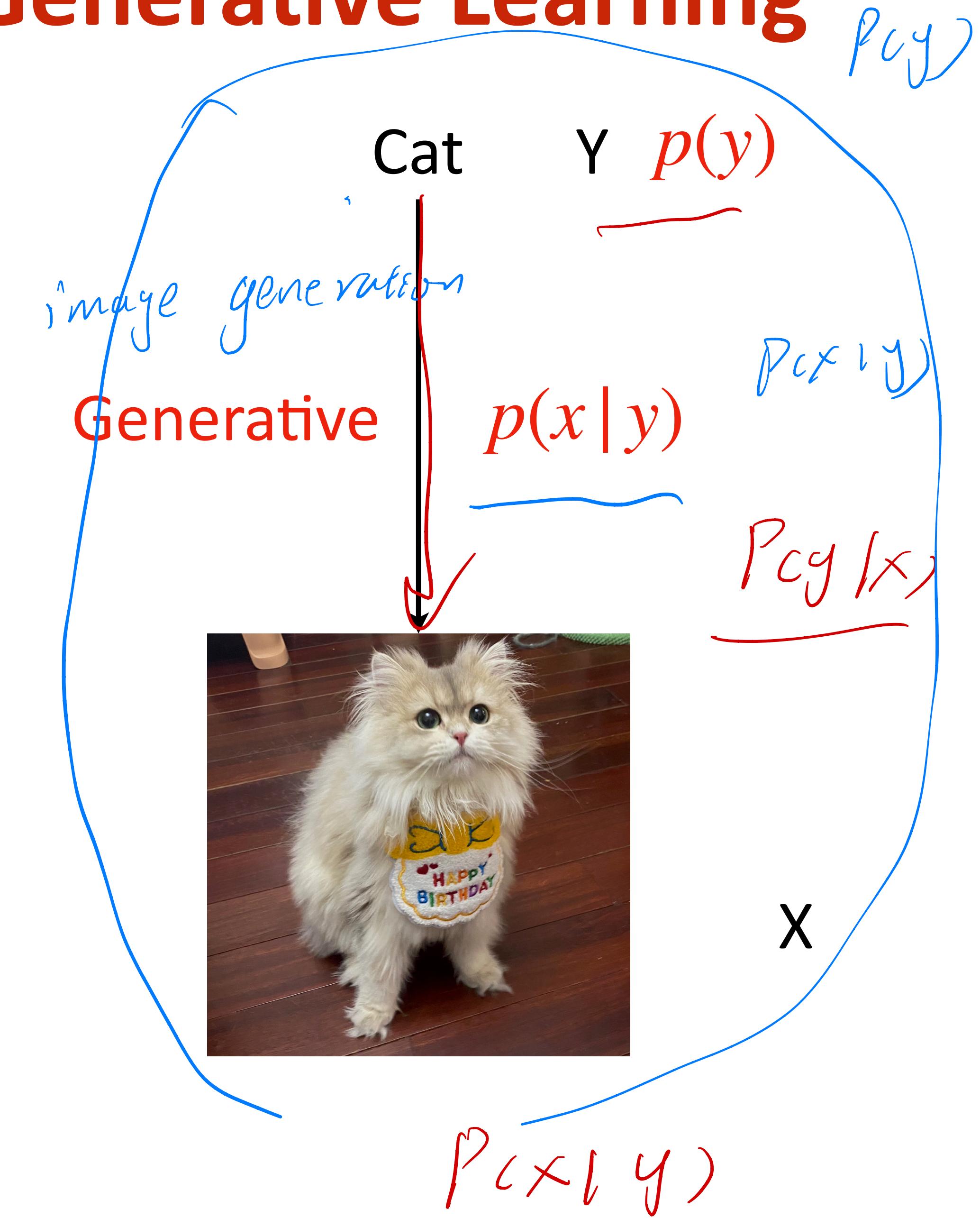
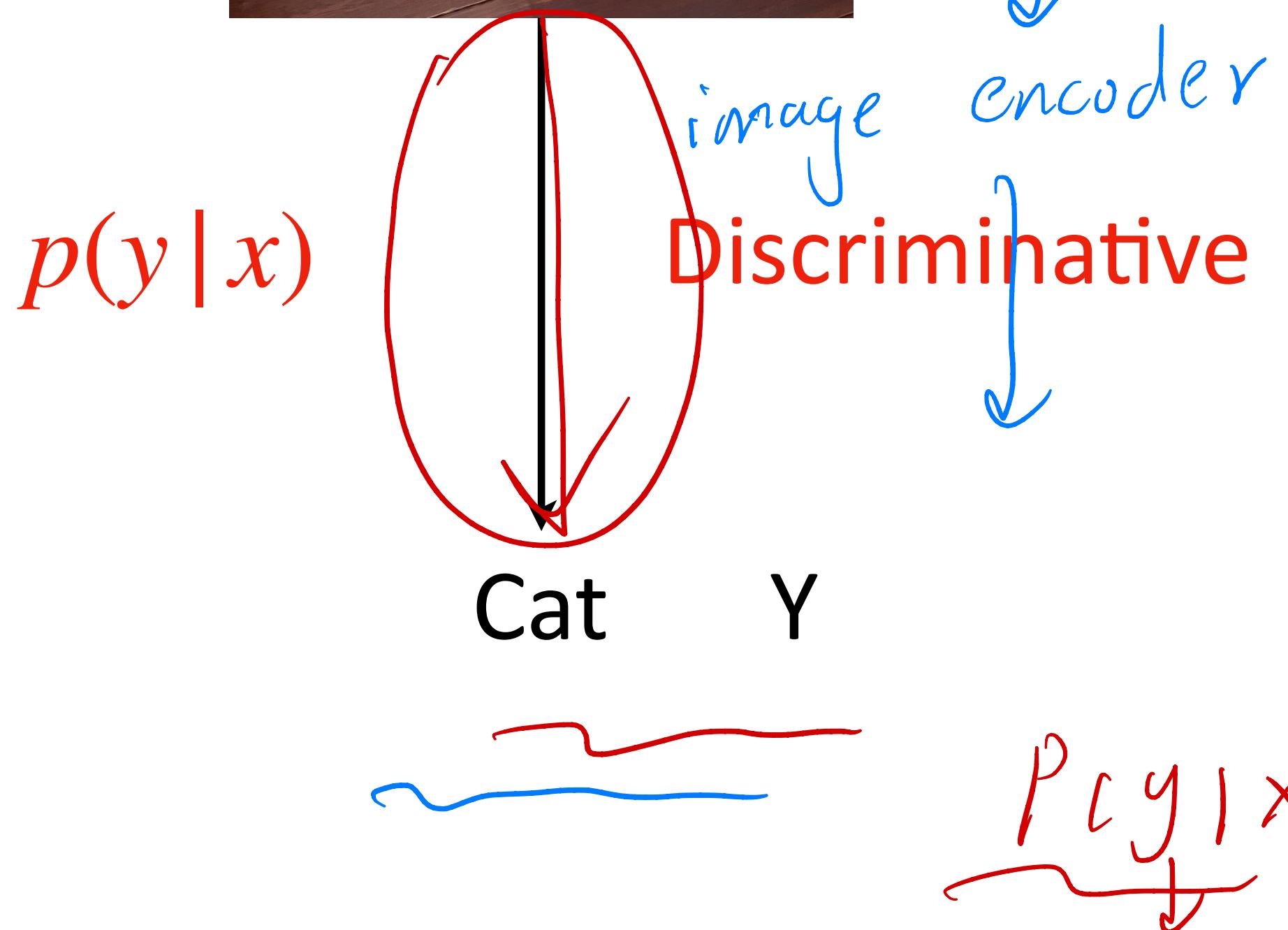


Video Generation Examples

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



Discriminative vs. Generative Learning



Bayes Rule

Bayes Rule

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

Bayes Rule

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

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

marginalization

$p(x, y) = \underbrace{p(x|y)}_{\text{conditional probability}} \underbrace{p(y)}_{\text{prior}} \quad p(y|x)$

$$P(y|x) = \frac{0.9}{P(x)}$$

$$\overbrace{P(x)}^{0.7}$$

$$\overbrace{P(x)}^{0.5}$$

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

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

$$\text{dog: } P(x|y)p(y) = 0.8$$

$$\text{mouse: } P(x|y)p(y) = 0.5$$

cat > dog > mouse

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

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

\times fixed label cat

y

$P(x)$

label dog

label mouse

$\text{proportional to } P(x|y)p(y)$

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

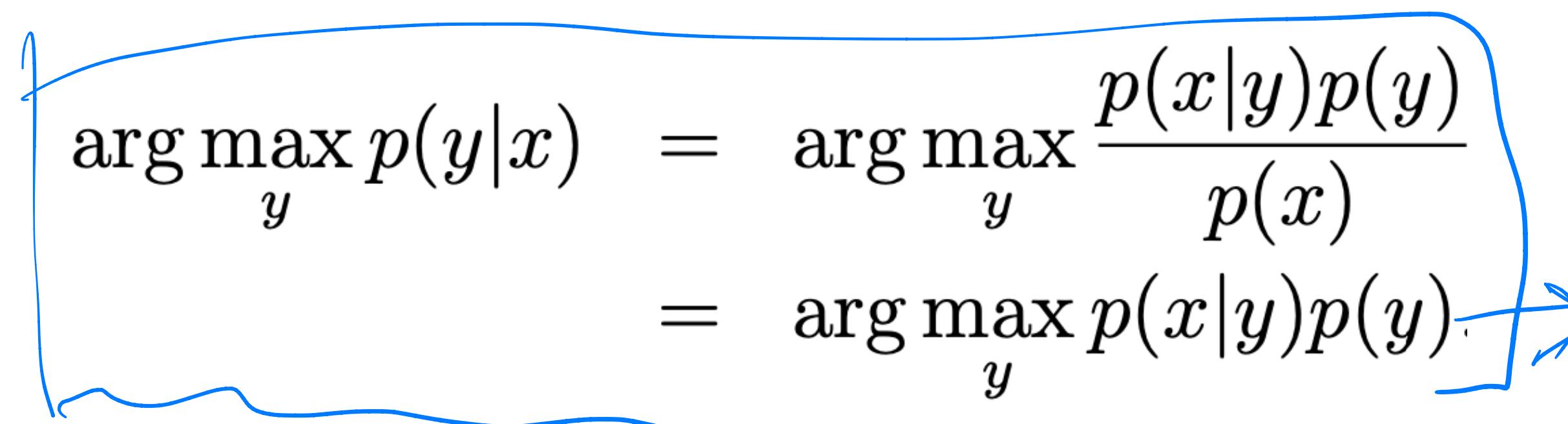
$\therefore P_{y=\text{dog}}(x) = \frac{P_{x|y=\text{dog}} P_{\text{dog}}}{P(x)}$

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:

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


Generative Models Compared to Discriminative Models

Pros:

Cons:

Generative Models Compared to Discriminative Models

Pros:

$$P(y) \quad P(x|y)$$
$$(x, y)$$
$$\boxed{P(x, y)} = P(y) P(x|y)$$

- Generative models can generate data (generation, data augmentation)

$$P(y|x)$$

$$P(x,y)$$
 cannot compute disrim.

$$(x, y)$$

$$(x, y)$$

Cons:

Generative Models Compared to Discriminative Models

Pros:



0.5

0.5

0.6 0.4

unsuccessful



- Generative models can generate data (generation, data augmentation)

- Inject prior information through the prior distribution

$P(y)$

cat

0.8

dog

0.1

mouse

0.1

$P(y)$

$P(x|y)$

0.5 0.5

$(0.6 \quad 0.4)$

Cons:

training data is not large

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~~

Cons:

may be learned in an unsupervised way when y is not available

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

x y

$P(x)$ language model

generative

P cyc IX

[Signature]

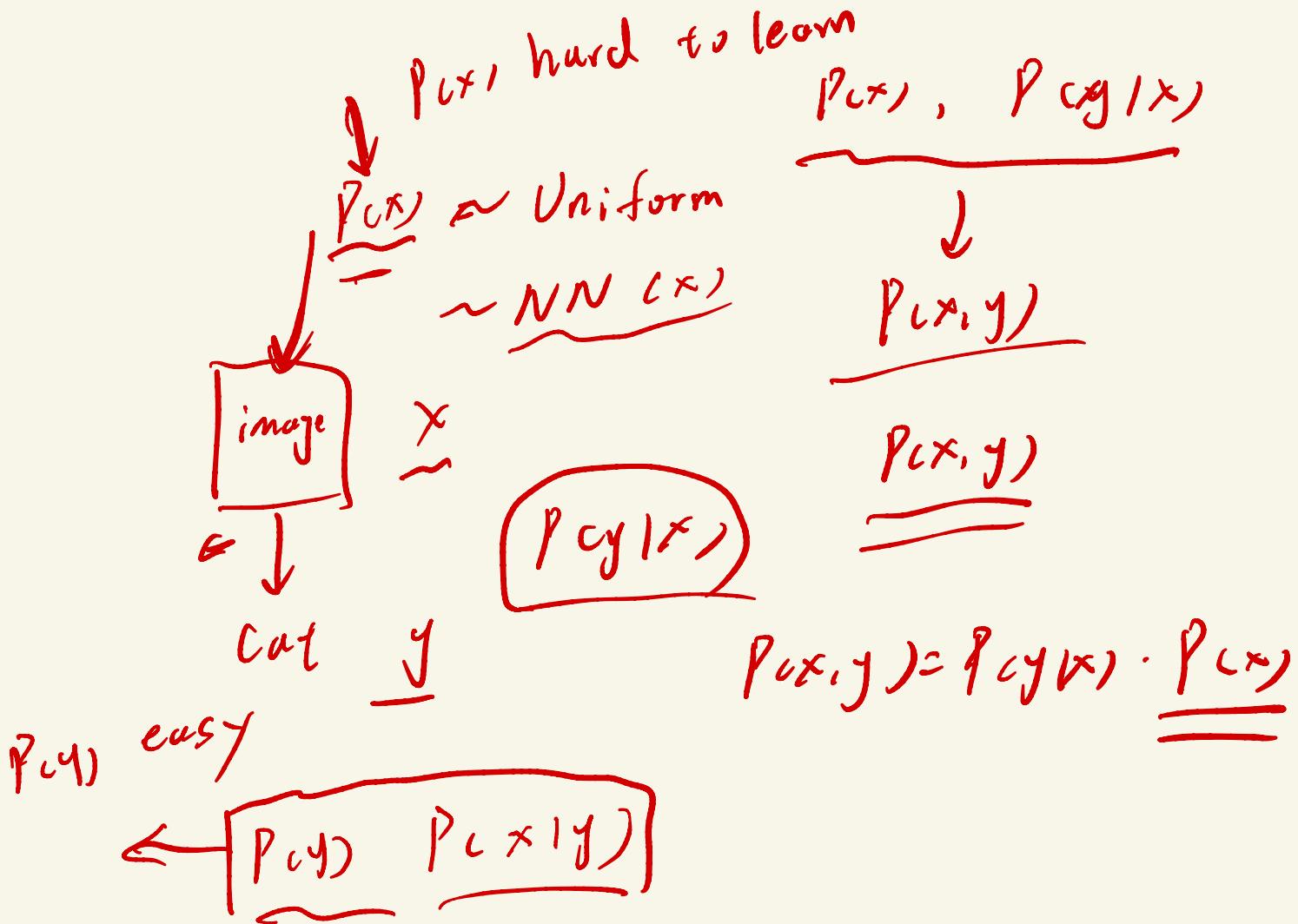
$P(x|y)$

~~Priority~~

$f(x, y)$

Fay

Poxy

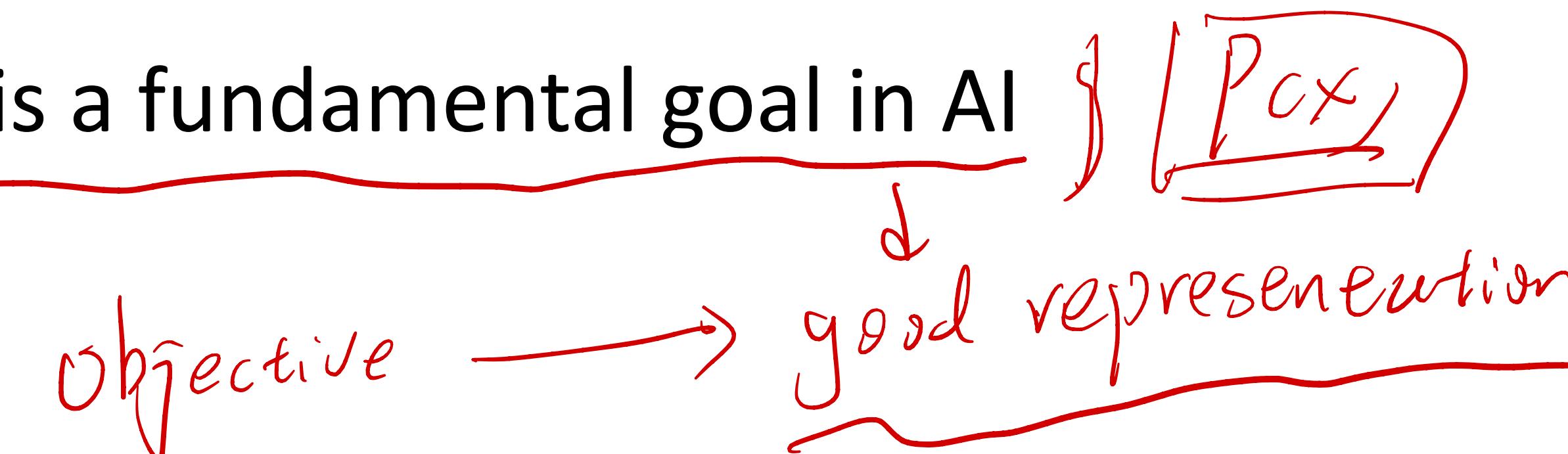


Generative Models Compared to Discriminative Models

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:



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

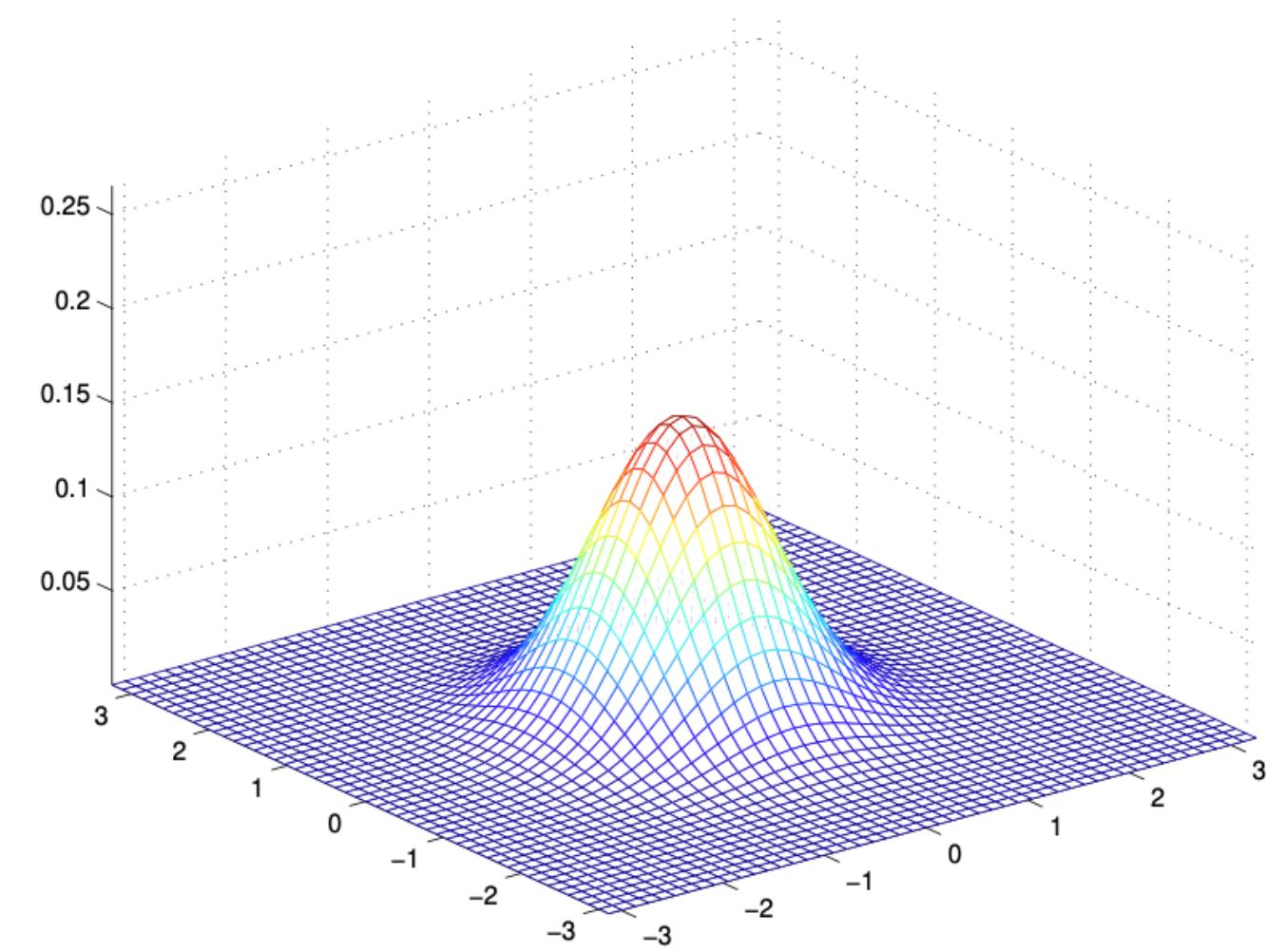
Gaussian Discriminant Analysis Model (GDA)

Multivariate Gaussian distribution

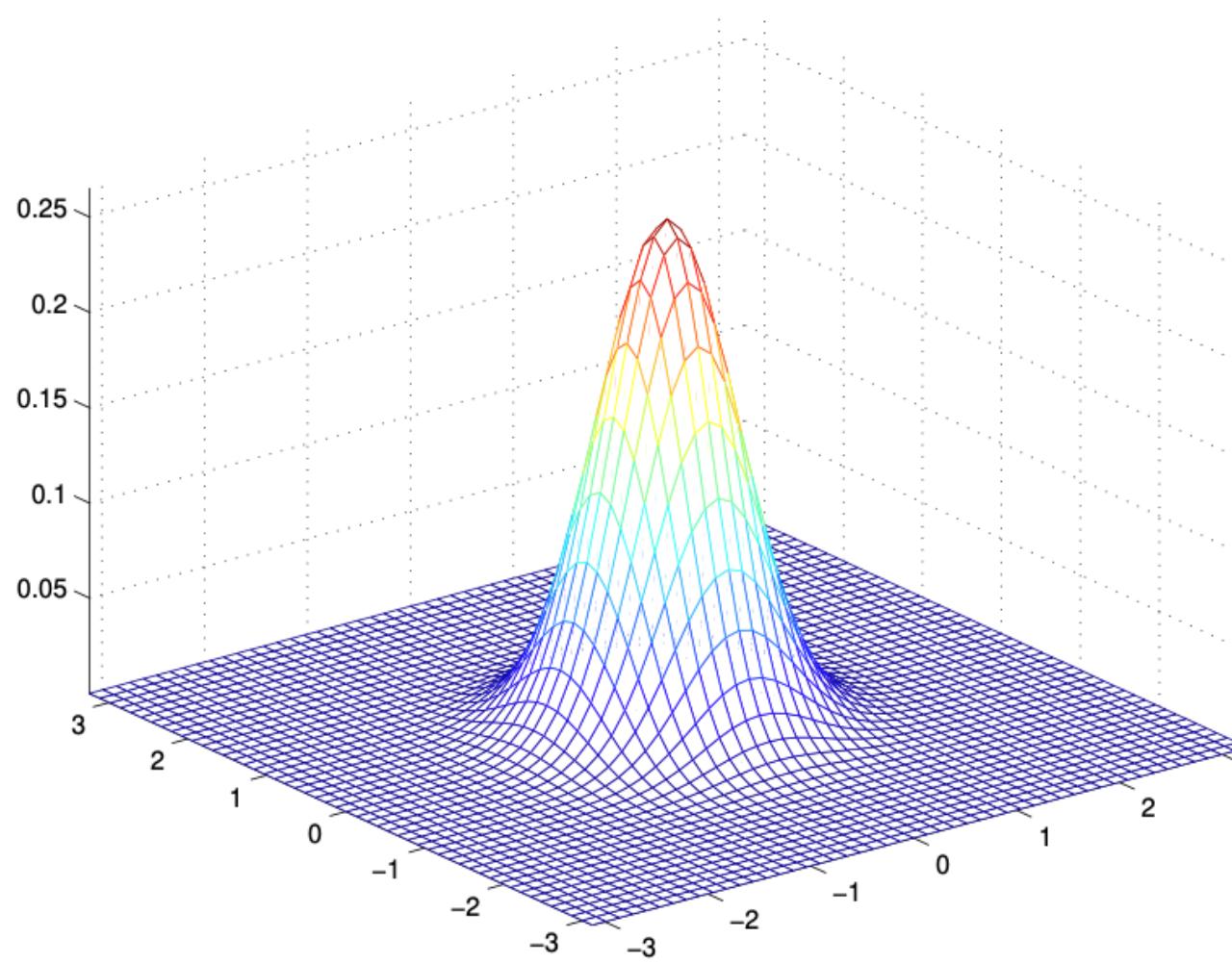
$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp \left(-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right)$$

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

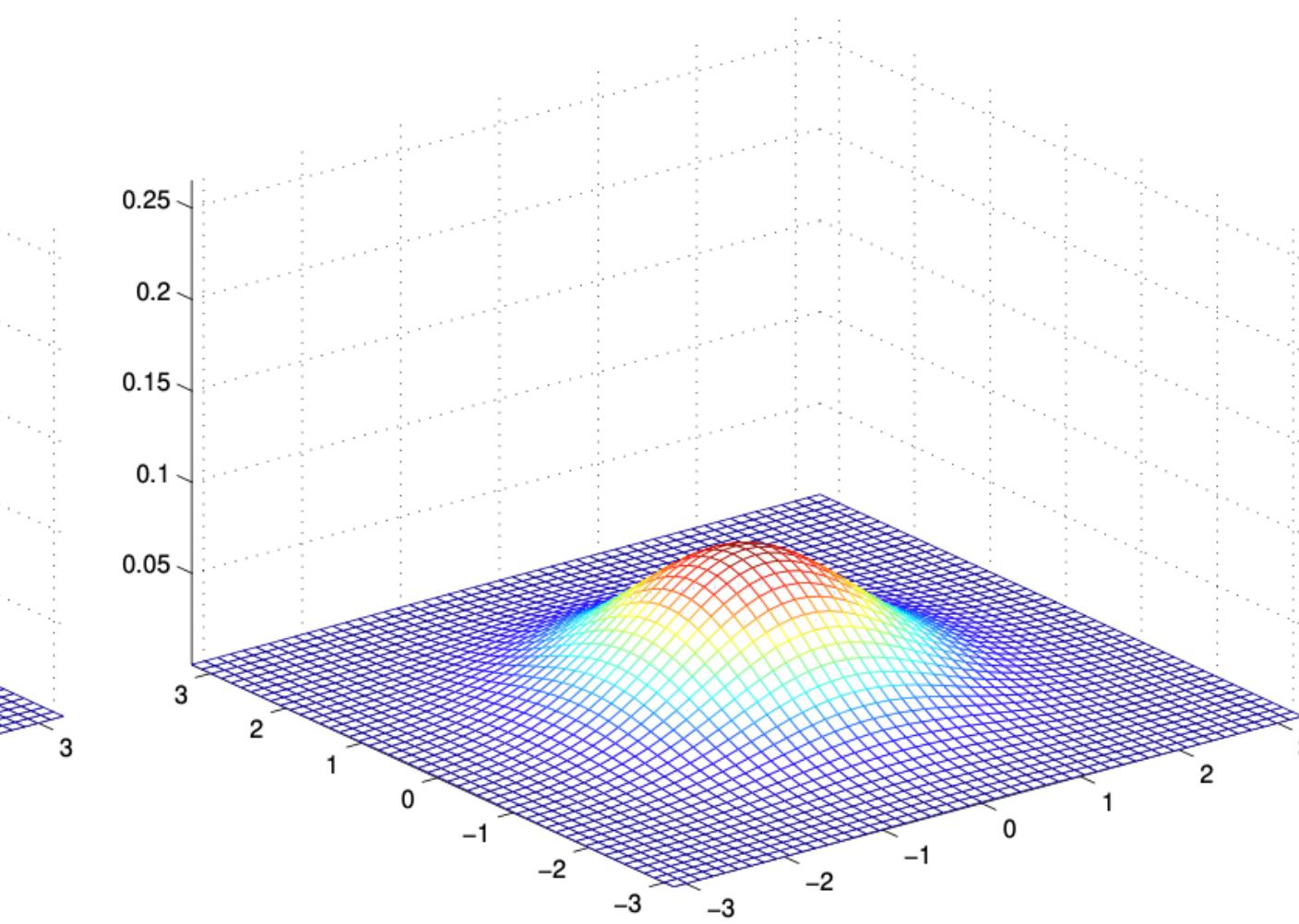
Examples of Multivariate Gaussian



$$\Sigma = I$$



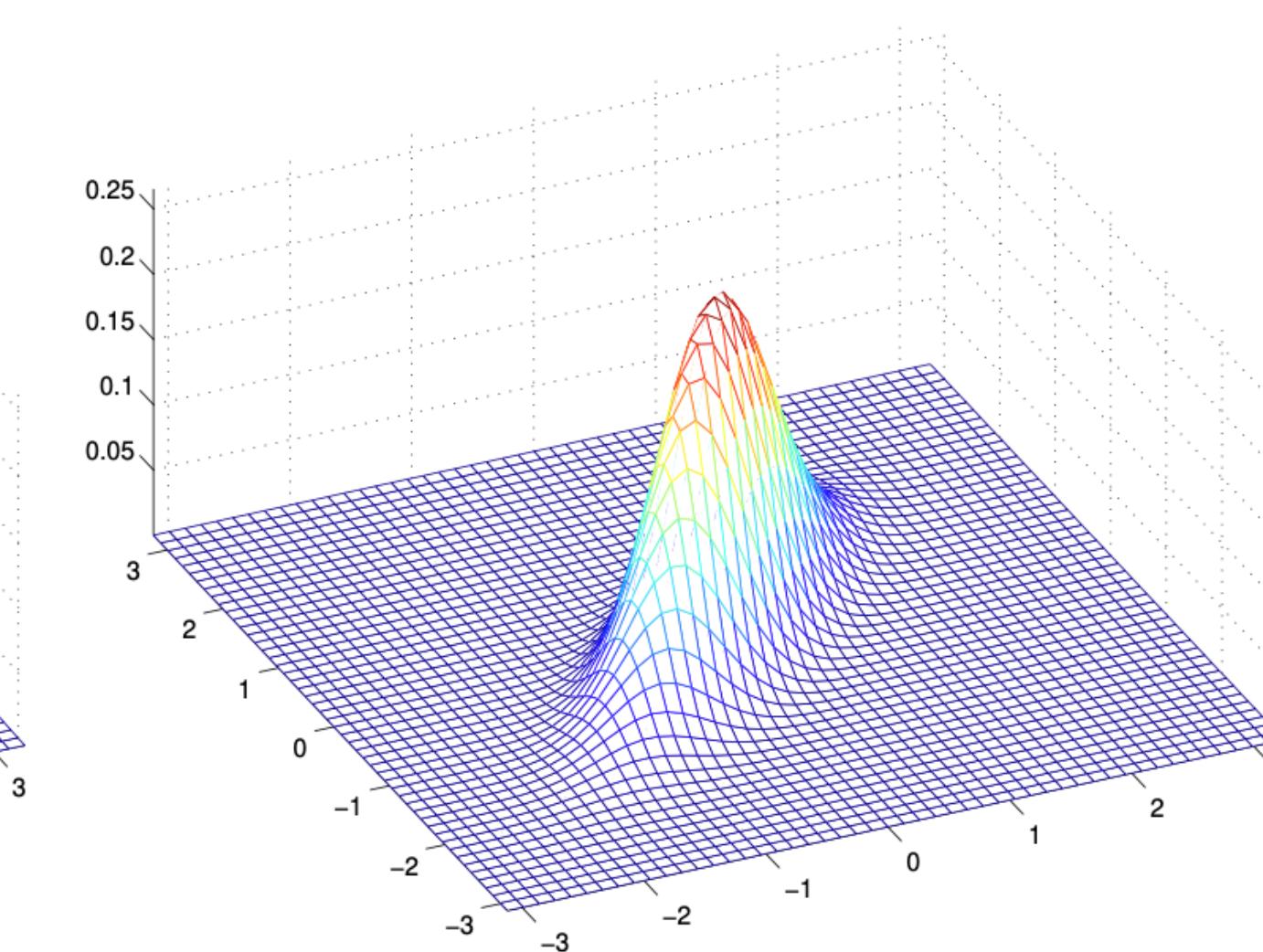
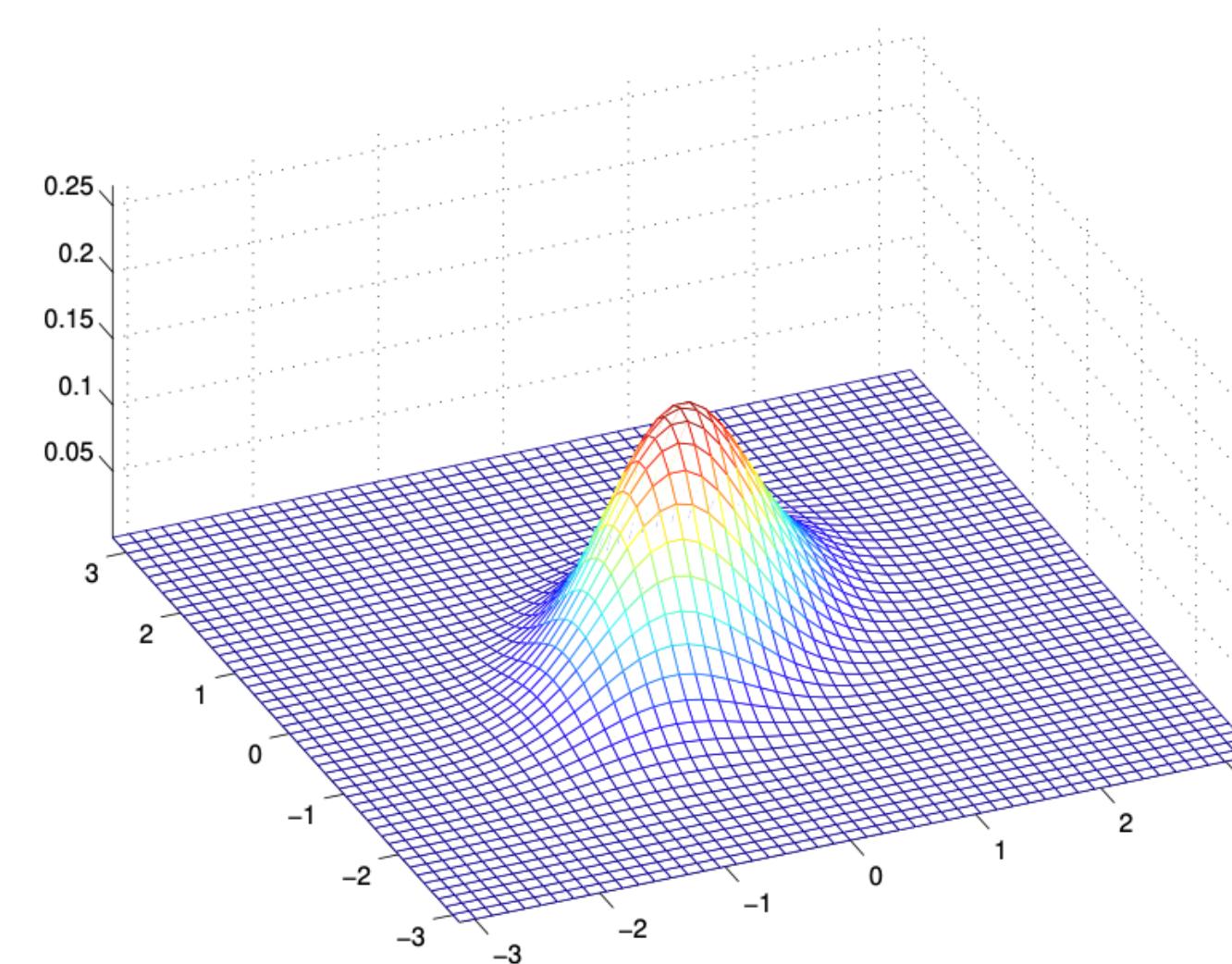
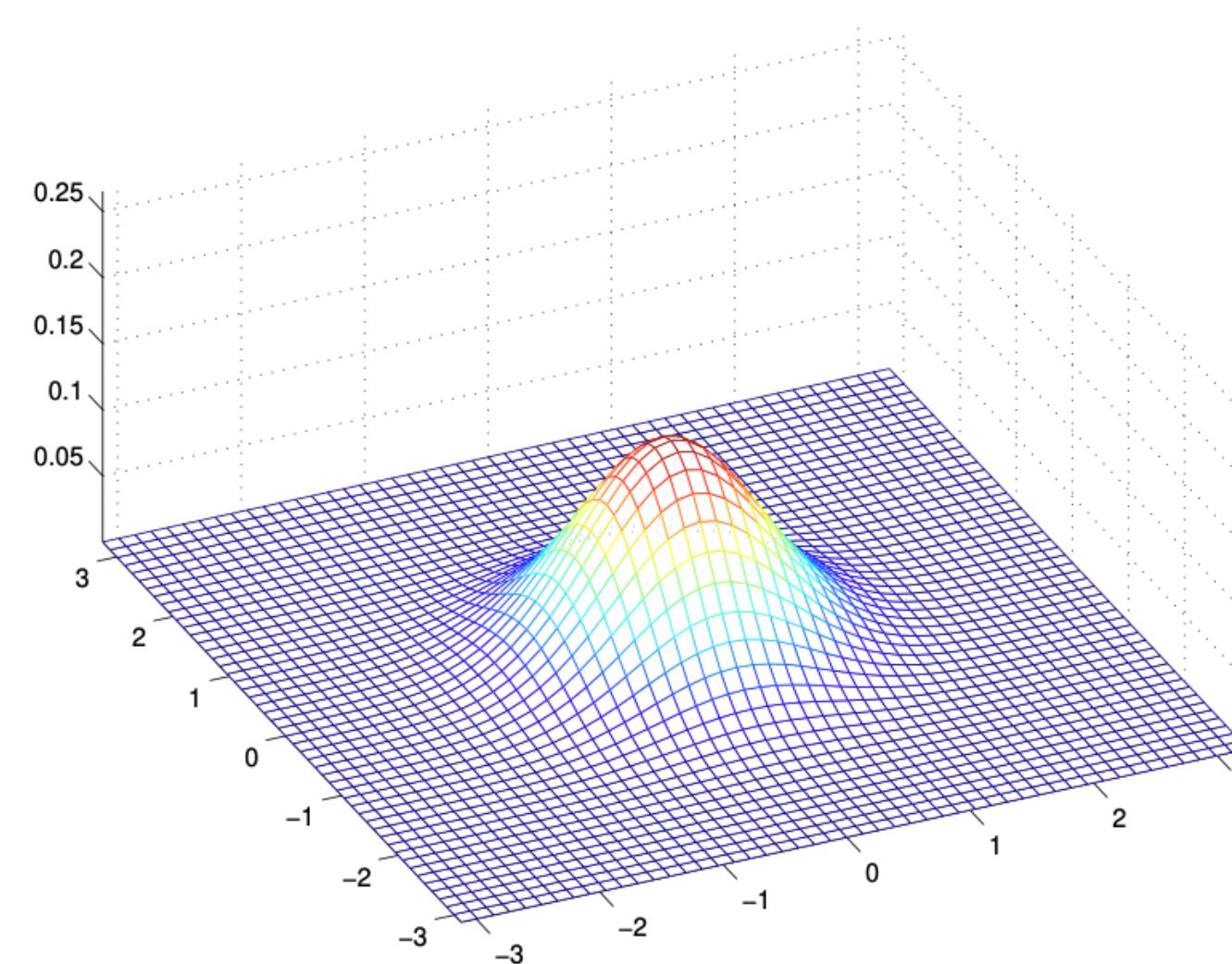
$$\Sigma = 0.6I$$



$$\Sigma = 2I$$



Examples of Multivariate Gaussian



$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

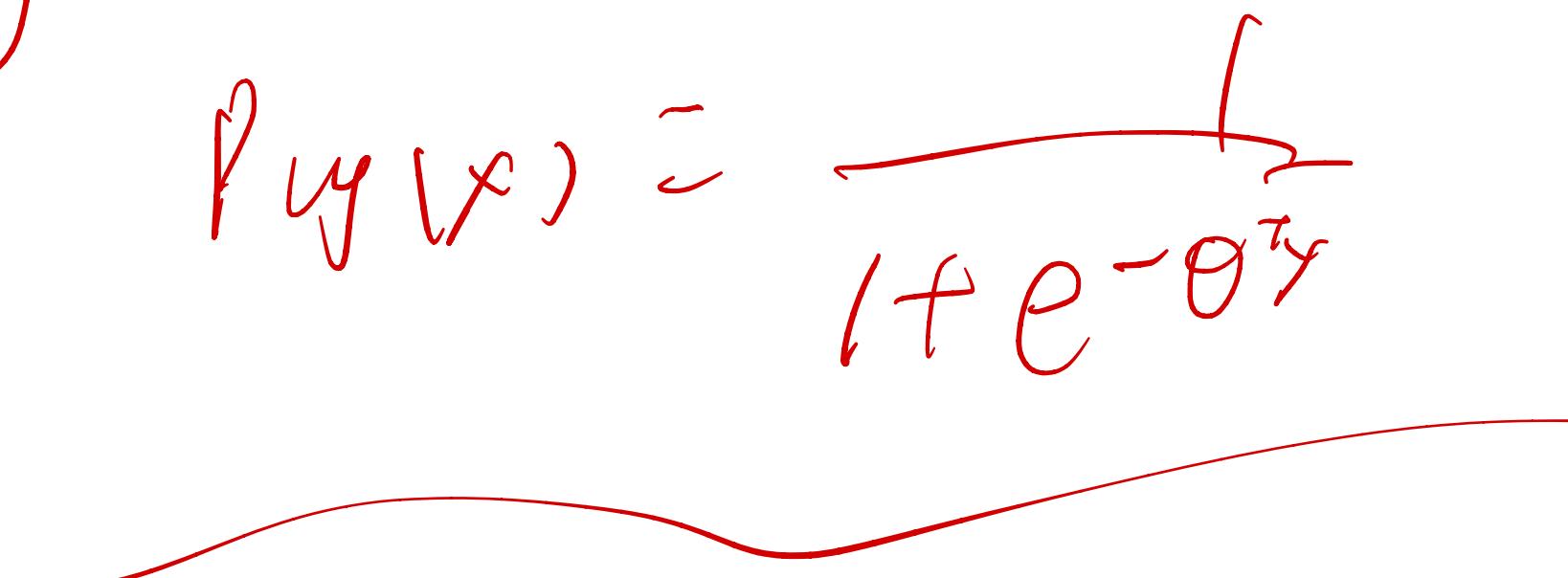
$$\Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

Gaussian Discriminant Analysis Model

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

y x logistic regression

$$P(y|x) = \frac{1}{1 + e^{-\theta^T x}}$$


Gaussian Discriminant Analysis Model

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

Assumption

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

$$P(y=1) = \phi \quad P(y=0) = 1 - \phi$$

$$P(x|y) = \mathcal{N}(\mu_y, \Sigma)$$

Gaussian Discriminant Analysis Model

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

Assumption

$$y \sim \text{Bernoulli}(\phi)$$

$$x|y=0 \sim \mathcal{N}(\mu_0, \Sigma)$$

$$x|y=1 \sim \mathcal{N}(\mu_1, \Sigma)$$

$$p(y) = \phi^y(1 - \phi)^{1-y}$$

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

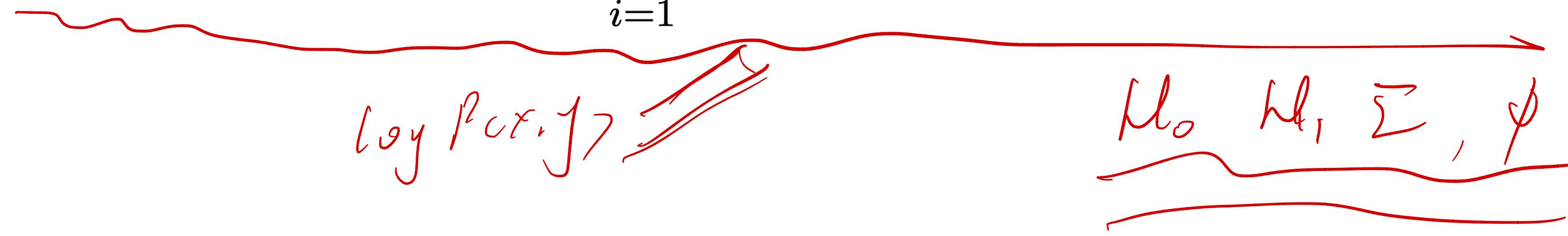
$$p(x|y=1) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_1)^T \Sigma^{-1} (x - \mu_1)\right)$$

$P(x, y)$

Maximum Likelihood Estimation

Maximum Likelihood Estimation

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



Maximum Likelihood Estimation

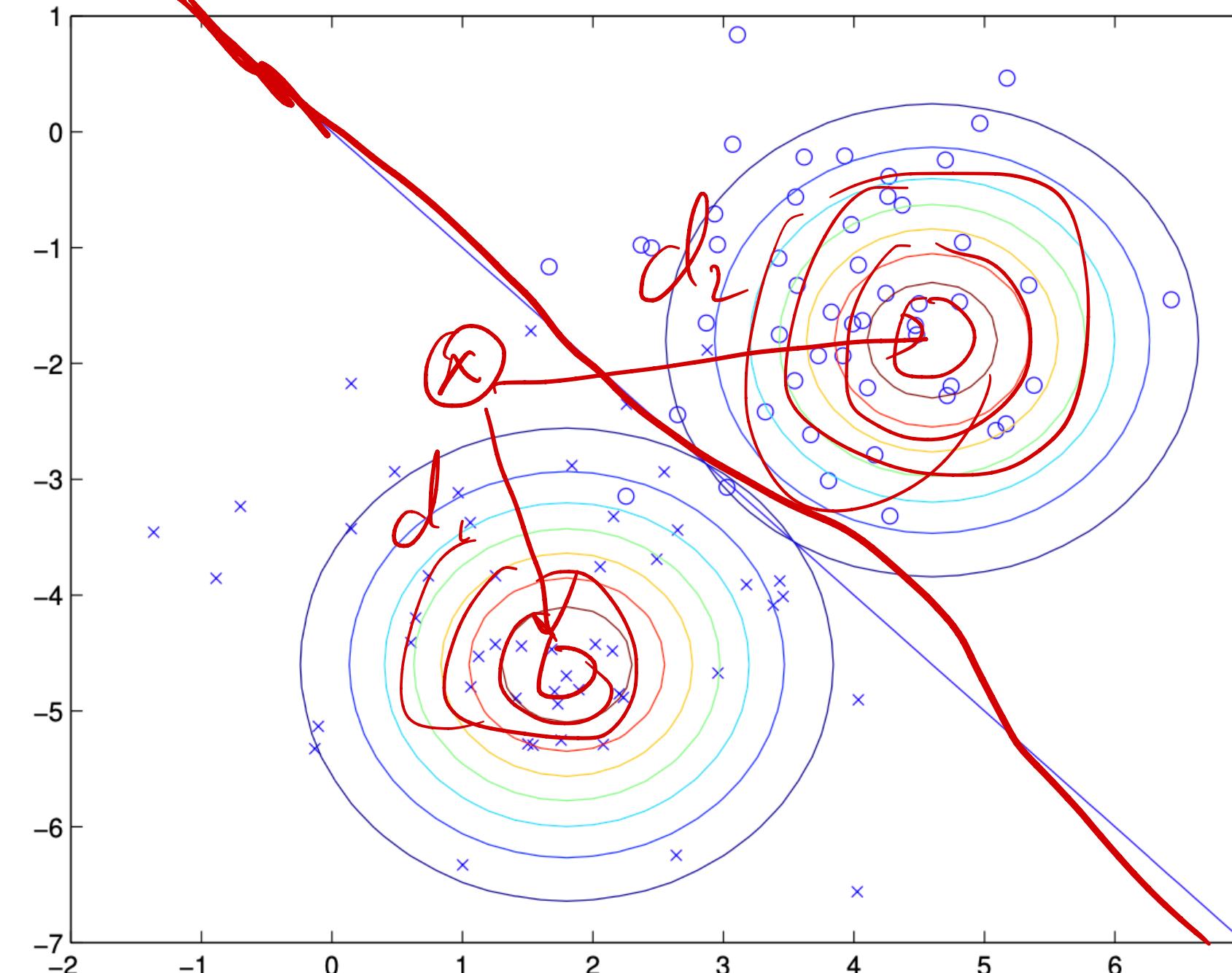
$$\begin{aligned}\ell(\phi, \mu_0, \mu_1, \Sigma) &= \log \prod_{i=1}^n p(x^{(i)}, y^{(i)}; \phi, \mu_0, \mu_1, \Sigma) \\ &= \log \prod_{i=1}^n p(x^{(i)}|y^{(i)}; \mu_0, \mu_1, \Sigma) p(y^{(i)}; \phi).\end{aligned}$$

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

Maximum Likelihood Estimation

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

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

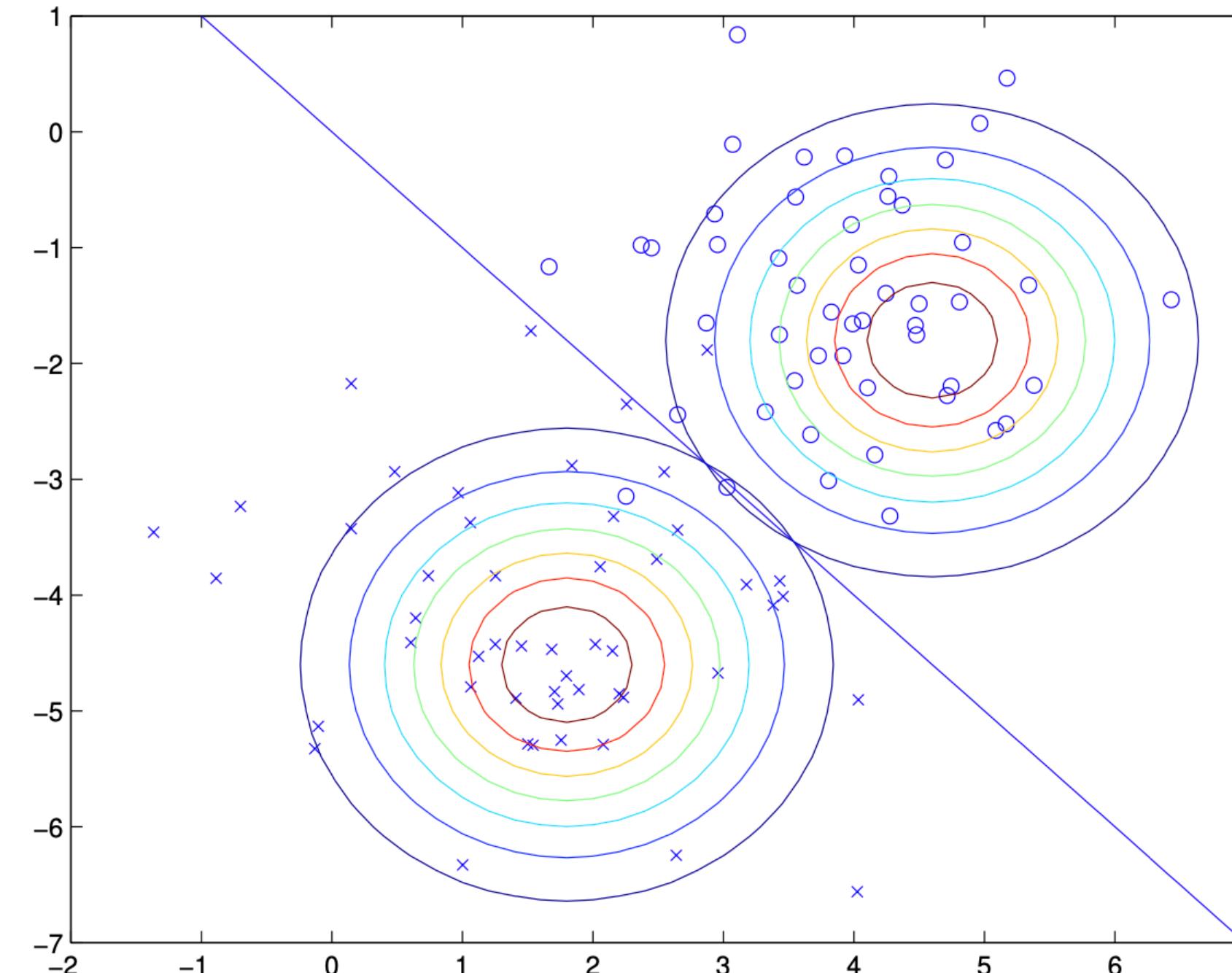


Maximum Likelihood Estimation

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

Why is the decision boundary linear?

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



$$P(y|x) \propto P(y) P(x|y)$$

$$P(y=1|x) = \left(\log \phi - \frac{1}{2} (x - \mu_1)^T \Sigma^{-1} (x - \mu_1) \right) / \frac{2}{\sigma^2}$$

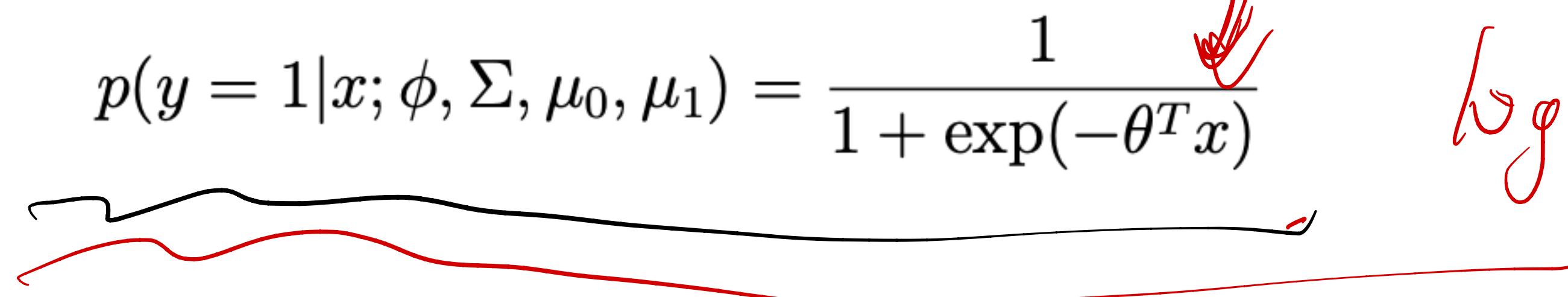
$$P(y=0|x) = \left(\log (1-\phi) - \frac{1}{2} (x - \mu_0)^T \Sigma^{-1} (x - \mu_0) \right) / \frac{2}{\sigma^2}$$

$$P(y=0|x) = P(y=1|x) \boxed{x^T \Sigma^{-1} x}$$

Connection Between GDA and Logistic Regression

Connection Between GDA and Logistic Regression

Through Bayes rule, we can show that

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


logistic regress

Connection Between GDA and Logistic Regression

Through Bayes rule, we can show that

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

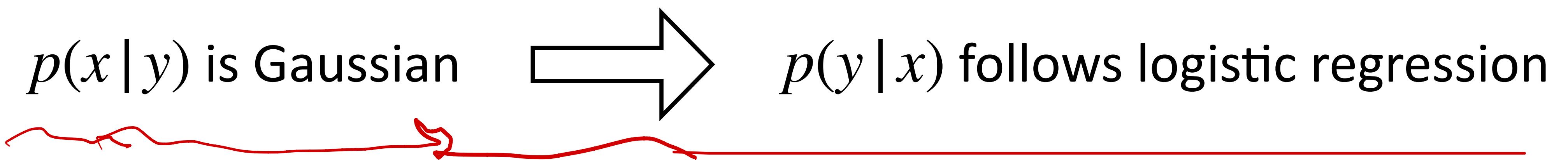
$$\underbrace{\theta = f(\phi, \Sigma, \mu_0, \mu_1)}$$

Connection Between GDA and Logistic Regression

Through Bayes rule, we can show that

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

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



Connection Between GDA and Logistic Regression

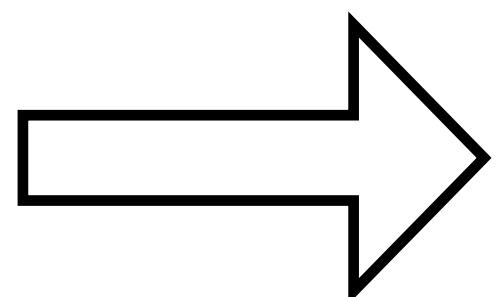
Through Bayes rule, we can show that

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

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

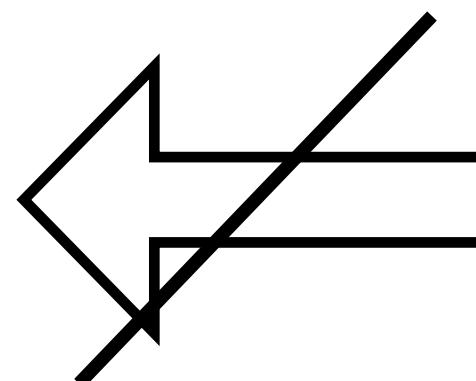
GDA is a subset of logistic regression family

$p(x|y)$ is Gaussian



$p(y|x)$ follows logistic regression

$p(x|y)$ is Gaussian



$p(y|x)$ follows logistic regression

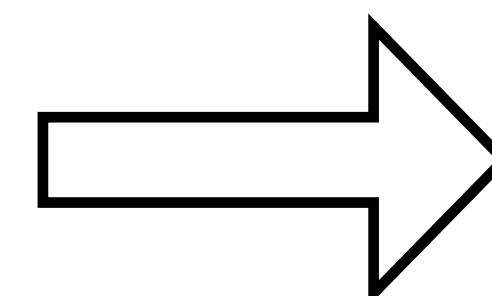
Connection Between GDA and Logistic Regression

Through Bayes rule, we can show that

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

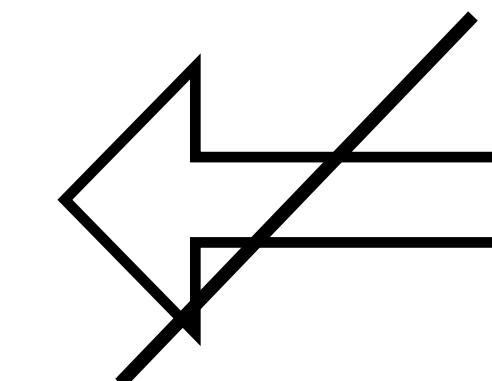
$$\theta = f(\phi, \Sigma, \mu_0, \mu_1)$$

$p(x|y)$ is Gaussian



$p(y|x)$ follows logistic regression

$p(x|y)$ is Gaussian



$p(y|x)$ follows logistic regression

Gaussian Discriminative Analysis model makes stronger assumptions

more powerful

Connection Between GDA and Logistic Regression

Connection Between GDA and Logistic Regression

Gaussian Discriminative Analysis (GDA) 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

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

freedom

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)

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

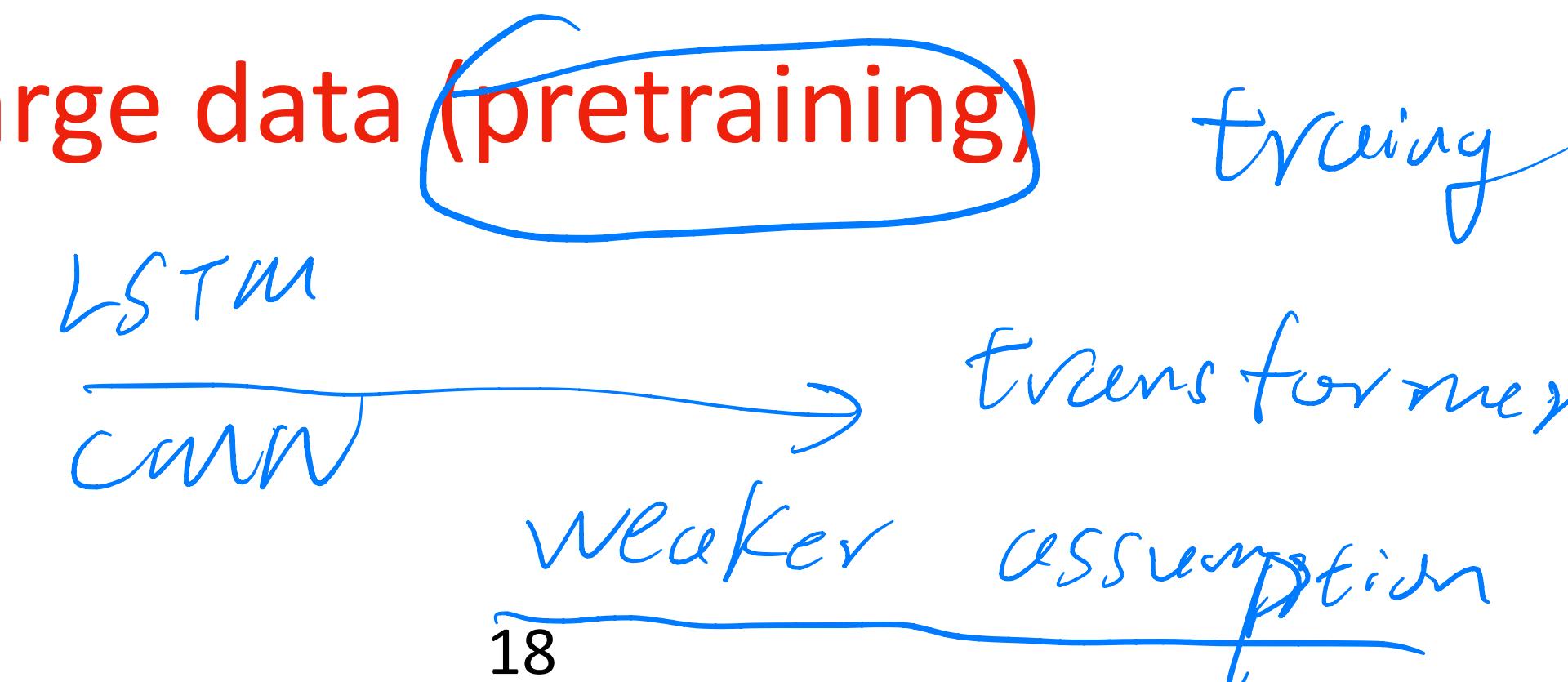
- When $x|y$ does not follow Gaussian in practice, GDA may or may not do well
- When $x|y$ does not follow Gaussian and the training data is large, the method that makes weaker assumptions (logistic regression) will always do better
- When $x|y$ indeed follows Gaussian and the training data is small,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

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) *training data is large*



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

The Bitter Lesson

<http://www.incompleteideas.net/IncIdeas/BitterLesson.html>

“The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin” — Rich Sutton

alpha go

Naive Bayes

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



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

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

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

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 $x_j = 0$

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 $x_j = 0$

$$x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \quad \left| \begin{array}{l} a \\ aardvark \\ aardwolf \\ \vdots \\ buy \\ \vdots \\ zygmurgy \end{array} \right. \quad \text{dictionary size}$$


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 $x_j = 0$

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

Dimension is the size of the dictionary

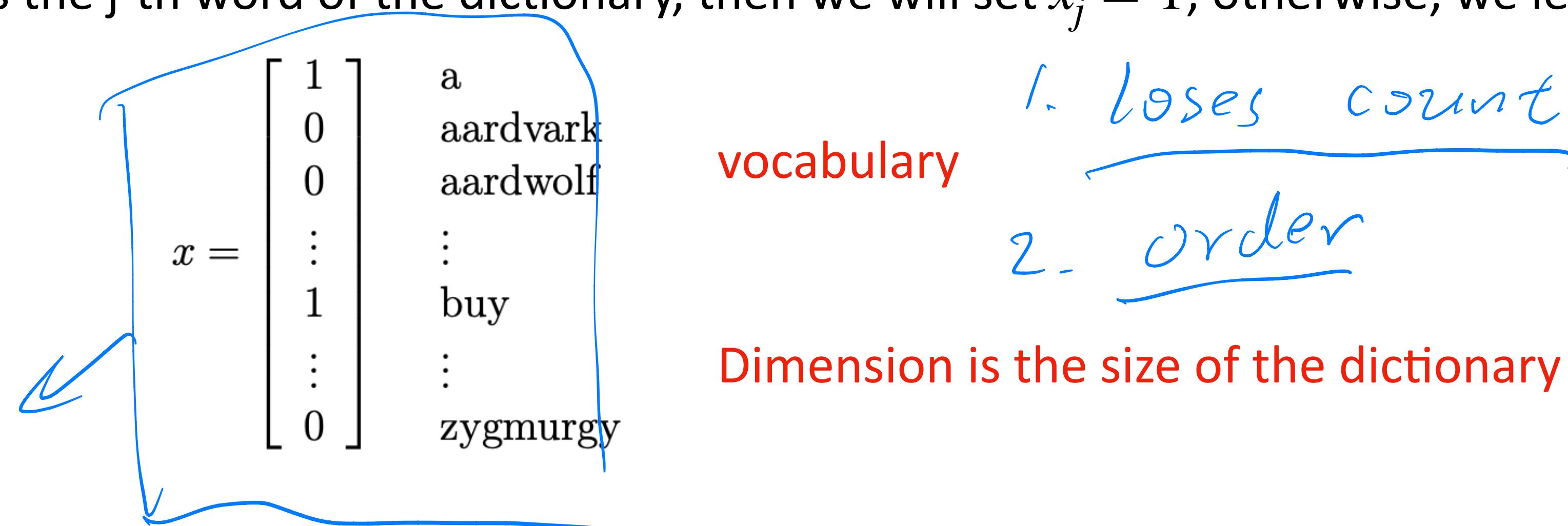
Naive Bayes

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 $x_j = 0$



Email Spam Classification

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

Email Spam Classification

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

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

$$\underbrace{P(x|y)}_{\text{Probability}} \left[\underbrace{\underbrace{2}_{\text{Number of words}}^{\text{50000}}}^{\text{Number of possible } x} \right]^{2^{50000}}$$

Email Spam Classification

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

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

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



Email Spam Classification

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

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 , $p(x_i | y) = p(x_i | y, x_j)$

$P_{C|X}$

Email Spam Classification

$$P(x_1, x_2, x_3 | y) = P(x_1 | y) P(x_2 | x_1, y) P(x_3 | x_1, x_2, y)$$

50000

Email Spam Classification

$$\begin{aligned} p(x_1, \dots, x_{50000} | y) & \xrightarrow{\text{Red oval}} \text{2 } 50000 \\ &= p(x_1 | y) p(x_2 | y, x_1) p(x_3 | y, x_1, x_2) \cdots p(x_{50000} | y, x_1, \dots, x_{49999}) \\ &= p(x_1 | y) p(x_2 | y) p(x_3 | y) \cdots p(x_{50000} | y) \\ &= \underbrace{\prod_{j=1}^d p(x_j | y)}_{\text{Blue bracket}} \quad | \\ &\quad \downarrow \end{aligned}$$

Email Spam Classification

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

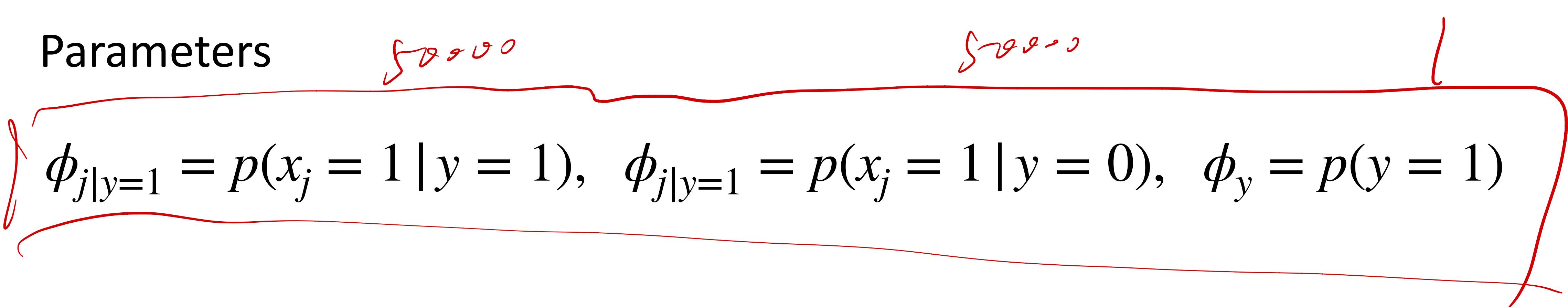
Email Spam Classification

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

Parameters

$\phi_{y|y=0}$

$\phi_{y|y=1}$



Email Spam Classification

$$p(x_1, \dots, x_{50000} | y) \quad \text{Autoregressive}$$

$$= p(x_1 | y) p(x_2 | y, x_1) p(x_3 | y, x_1, x_2) \cdots p(x_{50000} | y, x_1, \dots, x_{49999})$$

$$= p(x_1 | y) p(x_2 | y) p(x_3 | y) \cdots p(x_{50000} | y)$$

$$= \prod_{j=1}^d p(x_j | y)$$

$$2^{50000} \quad p(y) \quad P(x|y)$$

$$P(x, y)$$

Parameters

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

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

Maximum Likelihood Estimation

Maximum Likelihood Estimation

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

Maximum Likelihood Estimation

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

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

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

Prediction

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

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

Laplace Smoothing

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

Laplace Smoothing

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

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

Laplace Smoothing

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

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

Suppose the index in the dictionary for “learning” is q

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

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

Laplace Smoothing

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

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

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

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

Suppose the index in the dictionary for “learning” is q

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

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

Laplace Smoothing

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

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

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

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

Suppose the index in the dictionary for “learning” is q

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

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

Laplace Smoothing

Laplace Smoothing

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

Laplace Smoothing

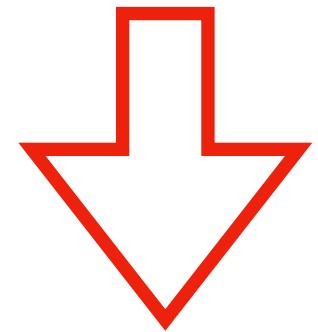
Take the problem of estimating the mean of a multinomial random variable z taking values in $\{1, \dots, k\}$. Given the independent observations $\{z^{(1)}, \dots, z^{(n)}\}$

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

Laplace Smoothing

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

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



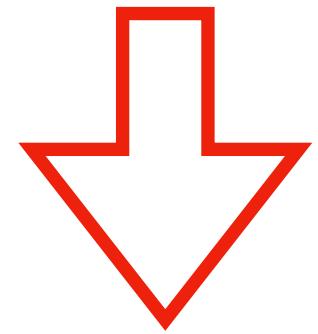
$$\phi_j = \frac{1 + \sum_{i=1}^n 1\{z^{(i)} = j\}}{k + n}$$

Laplace Smoothing

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

$$\phi_j = p(z = j)$$

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



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

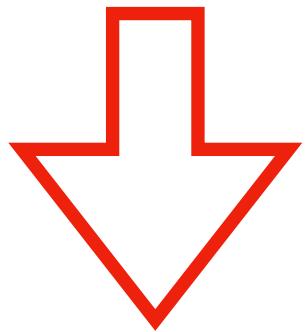
Why adding k to the denominator?

Laplace Smoothing

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

$$\phi_j = p(z = j)$$

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



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

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