Data Representation

Hao Dong

Peking University

Data Representation

p_{data}

$$\mathbf{x}^j \sim p_{data}$$
$$j = 1,2, \ldots |\mathcal{D}|$$

- dataset $\mathcal{D}$
- data distribution p_{data}
- model parameters $\theta \in \mathcal{M}$

- **How to represent (model) a data distribution?**

It can be an optimisation problem:

$$\min_{\theta \in \mathcal{M}} \mathcal{L}(p_{data}, p_\theta)$$

- **Why parametric models?**

They scale more efficiently with large dataset than non-parametric models.

Data Representation

p_{data}

$\mathbf{x}^j \sim p_{data}$
$j = 1,2, \dots |\mathcal{D}|$

- dataset $\mathcal{D}$
- data distribution p_{data}
- model parameters $\theta \in \mathcal{M}$

- **We want to learn a probability distribution $p(\mathrm{x})$ over $\mathbf{x}$**

1. **Generation (sampling):** $\mathbf{x}_{new} \sim p(\mathrm{x})$
2. **Density Estimation:** $p(\mathrm{x})$ high if $\mathbf{x}$ looks like a cat
3. **Unsupervised Representation Learning:**
   Discovering the underlying structure from the data distribution (e.g., ears, nose, eyes …)

Data Representation

- **Recap: Challenges from Lecture 1**

  - Representation ability

    How to represent $p(x)$

    For 1-D data x , the probability distribution $p(x)$ is simple, e.g., Gaussian?

    For high-dimensional data $\mathbf{x} = (x_1, x_2, \ldots, x_n)$,

    how do we learn the joint distribution $p(x_1, x_2, \ldots, x_n)$?

  - Learning method

    How do we measure and minimise the distance

    between the estimated distribution $p(x)$ (i.e., p_x) and the real distribution p_{data}?

    we can now perform generative process and density estimation

  - Inference

    How do we perform discriminative task?

    i.e., invert the generative process

Data Representation

How to do representation

- Problem of High-dimensional Data
- Less Parameters: Conditional Independence
- Less Parameters: Bayesian Network

How to do inference

- Naïve Bayes Classifier
- Discriminative vs. Generative Models
- Logistic Regression

How to be better

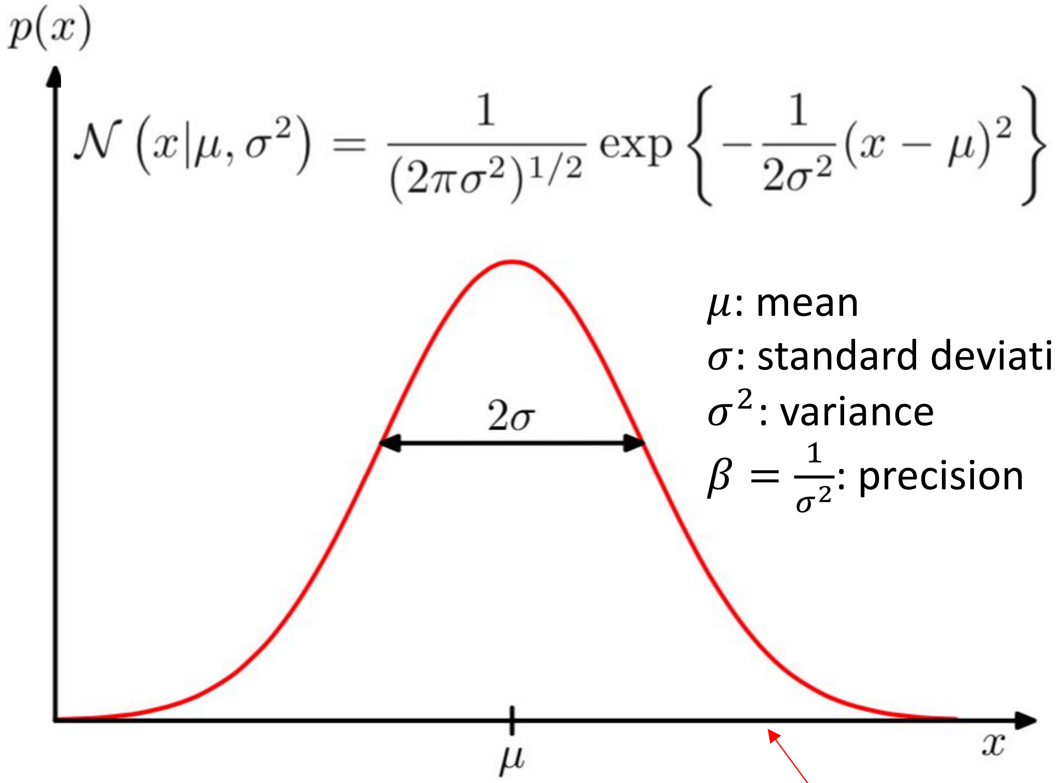
- Deep Neural Networks
- Continuous Variables

- Problem of High-dimensional Data
- Less Parameters: Conditional Independence
- Less Parameters: Bayesian Network
- Naïve Bayes Classifier
- Discriminative vs. Generative Models
- Logistic Regression
- Deep Neural Networks
- Continuous Variables

Problem of High-dimensional Data

- How to represent the height distribution (age from 30cm to 230cm)

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



μ: mean
σ: standard deviation
σ^2: variance
$\beta = \frac{1}{\sigma^2}$: precision

The probability of x to be this value

height

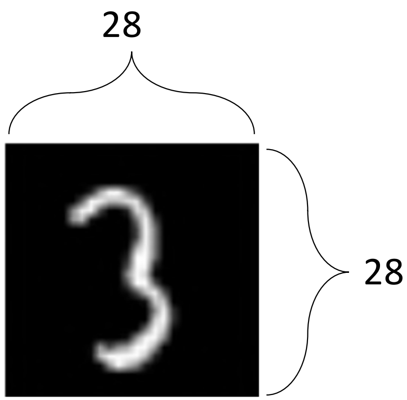
In this case, we have 200 states
and we need 2 parameters to represent
the probability distribution $p(x)$

μ, σ

Problem of High-dimensional Data

- How to represent a high-dimensional data $\mathbf{x} = (x_1, x_2, x_3, \ldots, x_n)$

MNIST dataset



784 random binary variables

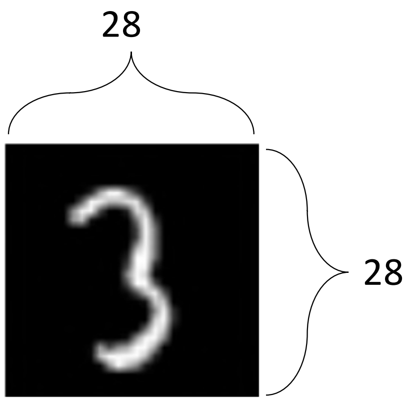
In MNIST, an images have $28 * 28 * 1 = 784$ binary values

So… how to represent $p(x_1, x_2, \ldots, x_{784})$?
how many number of parameters?

example from Stanford “deep generative models”

Problem of High-dimensional Data

- How to represent a high-dimensional data $\mathbf{x} = (x_1, x_2, x_3, \dots, x_n)$



(Bernoulli random variables)

As x can be either 0 or 1, i.e., only 2 states

(Joint distribution)

The number of possible state for $p(x_1, x_2, \dots, x_n)$ is $\mathbf{2^n}$

which is far larger than the number of data sample

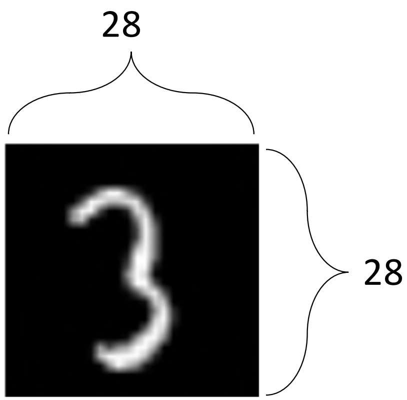
We need a super-large memory to store $p(x_1, x_2, \dots, x_n)$

even we have such large memory, we do not have enough data

to learn/model it

Problem of High-dimensional Data

- How to represent a high-dimensional data $\mathbf{x} = (x_1, x_2, x_3, \dots, x_n)$



784 random binary variables

$p(x_1, x_2, \dots, x_n)$ has $\mathbf{2^n}$ states, then …

How many number of parameters to model $p(x_1, x_2, \dots, x_n)$? ： $\mathbf{2^n}$ -1?

Recap: **Product Rule**

$$p(x_1, x_2) = p(x_1)p(x_2|x_1)$$

$$p(x_1, x_2, x_3) = p(x_1, x_2)p(x_3|x_1, x_2) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)$$

…

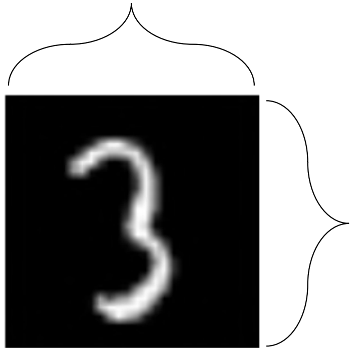
$$\boxed{p(x_1, x_2, \dots, x_n) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2) \dots p(x_n|x_1, \dots, x_{n-1})}$$

Problem of High-dimensional Data

- How to represent a high-dimensional data $\mathbf{x} = (x_1, x_2, x_3, \ldots, x_n)$

$$\boxed{p(x_1, x_2, \ldots, x_n) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2) \ldots p(x_n|x_1, \ldots, x_{n-1})}$$

28



28

784 random binary variables

- $p(x_1)$ need 1 parameter, the probability of x_1 to be 1 (as it is a binary variable)
- $p(x_2|x_1)$ need 2 parameters, i.e., $p(x_2|x_1 = 0)$ and $p(x_2|x_1 = 1)$
- $p(x_3|x_1, x_2)$ need 4 parameters, i.e., $p(x_3|x_1 = 0, x_2 = 0)$, $p(x_3|x_1 = 0, x_2 = 1)$
  $p(x_3|x_1 = 1, x_2 = 0)$, $p(x_3|x_1 = 1, x_2 = 1)$

So ... The number of parameters to model $p(x_1, x_2, \ldots, x_n)$ is:

$$\boxed{1 + 2 + 4 + \cdots + 2^{n-1} = 2^n - 1}$$

(when variables are binary)

Problem of High-dimensional Data

- How to represent a high-dimensional data $\mathbf{x} = (x_1, x_2, x_3, \dots, x_n)$

28

28

784 random binary variables

Product Rule:

$$\underbrace{p(x_1, x_2, \dots, x_n)}_{2^n \text{ states}} = \underbrace{p(x_1)p(x_2|x_1)p(x_3|x_1, x_2) \dots p(x_n|x_1, \dots, x_{n-1})}_{2^n - 1 \text{ parameters}}$$

$2^n - 1$ is exponential,

the product rule does not help to reduce the num of parameters

Problem of High-dimensional Data

- How to represent a high-dimensional data $\mathbf{x} = (x_1, x_2, x_3, \dots, x_n)$

In practice

1) The x can be continuous, i.e., infinite states

2) The number of x can be millions

For simplicity

We use binary x and MNIST for demo

北京大學 PEKING UNIVERSITY

- Problem of High-dimensional Data
- Less Parameters: Conditional Independence
- Less Parameters: Bayesian Network
- Naïve Bayes Classifier
- Discriminative vs. Generative Models
- Logistic Regression
- Deep Neural Networks
- Continuous Variables

Less Parameters: Conditional Independence

- How to reduce the number of parameter to represent $p(x_1, x_2, \ldots, x_n)$?

Product Rule does not help:

$$\underbrace{p(x_1, x_2, \ldots, x_n)}_{2^n \text{ states}} = \underbrace{p(x_1)p(x_2|x_1)p(x_3|x_1, x_2) \ldots p(x_n|x_1, \ldots, x_{n-1})}_{2^n - 1 \text{ parameters}}$$

28

28

784 random binary variables

Less Parameters: Conditional Independence

- How to reduce the number of parameter to represent $p(x_1, x_2, \ldots, x_n)$?

Recap: If variables x_1, x_2 are conditional independent given variable x_3, denotes as $x_1 \perp x_2 \mid x_3$

$$p(x_1, x_2|x_3) = p(x_1|x_3)p(x_2|x_3)$$

if not independent:

$$p(x_1, x_2|x_3) = \frac{p(x_1,x_2,x_3)}{p(x_3)} = \frac{p(x_1,x_2,x_3)}{p(x_2,x_3)}\frac{p(x_2,x_3)}{p(x_3)} = p(x_1|x_2, x_3)p(x_2|x_3)$$

so we can have $p(x_1|x_2, x_3)p(x_2|x_3) = p(x_1|x_3)p(x_2|x_3)$ **if** $x_1 \perp x_2 \mid x_3$

Less Parameters: Conditional Independence

- How to reduce the number of parameter to represent $p(x_1, x_2, \dots, x_n)$?

Given product rule: $p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)p(x_4|x_1, x_2, x_3)$

If $x_4 \perp x_2 \mid \{x_1, x_3\}$, we can simplify it as:

$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)p(x_4|x_1, \cancel{x_2}, x_3)$$

If $x_2 \perp \{x_1, x_3\} \mid x_4$, we can simplify it as:

$$p(x_1, x_2, x_3, x_4) = p(x_4, x_3, x_2, x_1) = p(x_4)p(x_3|x_4)p(x_2|x_3, x_4)p(x_1|x_2, x_3, x_4)$$
$$= p(x_4)p(x_3|x_4)p(x_2|x_3, x_4)p(x_1|\cancel{x_2, x_3}, x_4)$$

Less Parameters: Conditional Independence

- How to reduce the number of parameter to represent $p(x_1, x_2, \dots, x_n)$?

In an **extreme case**, if $x_{i+1} \perp \{x_1, x_2 \dots x_{i-1}\} \mid x_i$,i.e., the next variable only related to the current variable (Markov model!)

$$\begin{aligned} p(x_1, x_2, x_3, x_4) &= p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)p(x_4|x_1, x_2, x_3) \\ &= p(x_1)p(x_2|x_1)p(x_3|\cancel{x_1}, x_2)p(x_4|\cancel{x_1, x_2}, x_3) \\ &= p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_3) \end{aligned}$$

If x are binary variables

$2n - 1$ parameters << $2^n - 1$ parameters

So …

if conditional independencies exist, the number of parameter can be reduced!!

Less Parameters: Conditional Independence

- How to reduce the number of parameter to represent $p(x_1, x_2, \dots, x_n)$?

In a **MORE extreme case**, if x_i are independent identical (IID)

$$\begin{aligned} p(x_1, x_2, x_3, x_4) &= p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)p(x_4|x_1, x_2, x_3) \\ &= p(x_1)p(x_2|\cancel{x_1})p(x_3|\cancel{x_1, x_2})p(x_4|\cancel{x_1, x_2, x_3}) \\ &= p(x_1)p(x_2)p(x_3)p(x_4) \end{aligned}$$

However, in practice, there exists “**relationship**” between variables

the independence assumption is not practical…

e.g., the following random samples would not happen

- Problem of High-dimensional Data
- Less Parameters: Conditional Independence
- Less Parameters: Bayesian Network
- Naïve Bayes Classifier
- Discriminative vs. Generative Models
- Logistic Regression
- Deep Neural Networks
- Continuous Variables

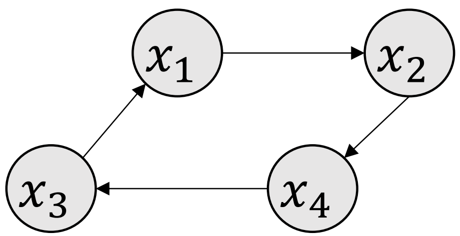
Less Parameters: Bayesian Network

- **Key idea:**

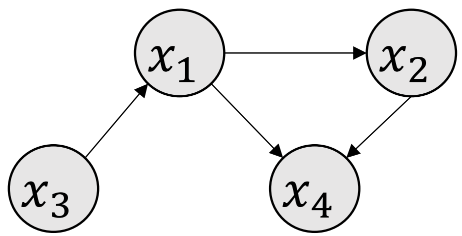
Joint distribution: $p(x_1, x_2, \ldots, x_n) = \underbrace{p(x_1)p(x_2|x_1)p(x_3|x_1, x_2) \ldots p(x_n|x_1, \ldots, x_{n-1})}_{2^n - 1 \text{ parameters if } x \text{ are binary variables}}$

use conditional distribution instead of joint distribution to reduce the num of parameters

Bayesian network structure is a **Directed Acyclic Graph**, $G = (V, E)$
where V means vertexes, E means edges



Directed Cycle



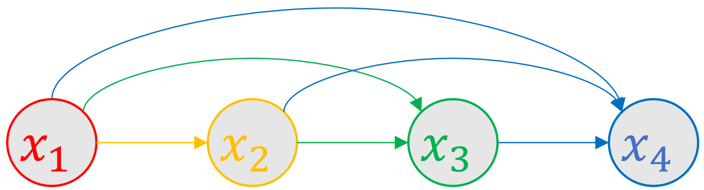
Directed Acyclic Graph

Less Parameters: Bayesian Network

- **Key idea:**

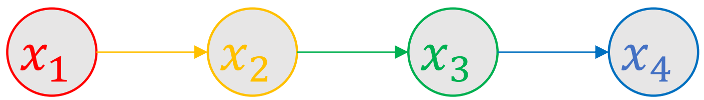
Bayesian network structure is a **Directed Acyclic Graph**, $G = (V, E)$

Joint distribution: $p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)p(x_4|x_1, x_2, x_3)$



If $x_{i+1} \perp \{x_1, x_2 \ldots x_{i-1}\} \mid x_i$

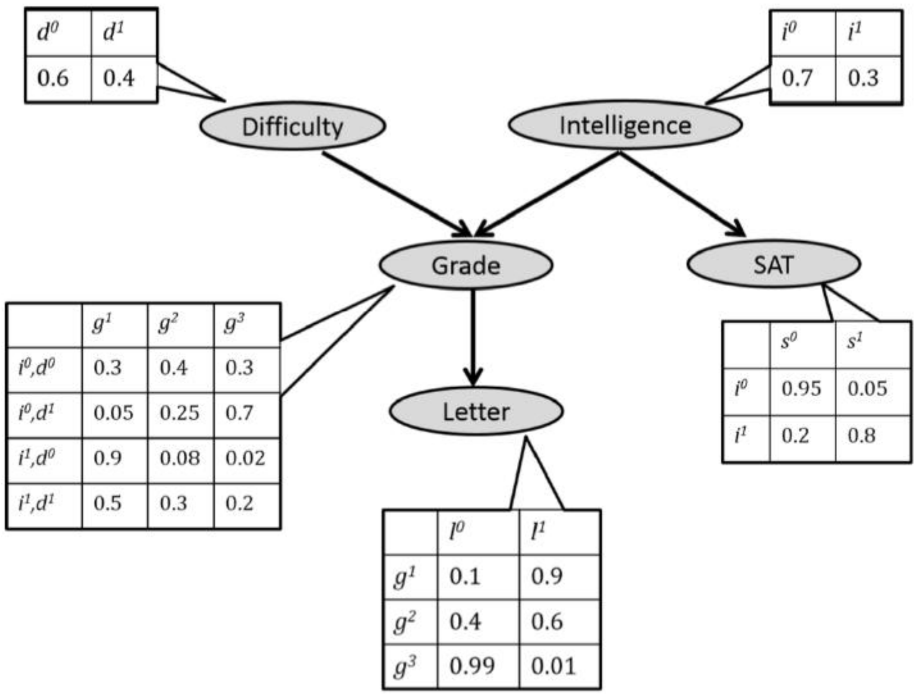
$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|\cancel{x_1}, x_2)p(x_4|\cancel{x_1, x_2}, x_3)$$
$$= p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_3)$$



Less edges == Less parameters

Less Parameters: Bayesian Network

- **Example**



$$p(d,i,g,s,l) = p(d)p(i|d)p(g|d,i)p(s|d,i,g)p(l|d,i,g,s)$$

According to the left Bayesian Net, we have the independencies:

$$d \perp i \qquad s \perp \{d,g\} \qquad l \perp \{d,i,s\}$$

So that ..

$$p(d,i,g,s,l) = p(d)p(i)p(g|i,d)p(s|i)p(l|g)$$

Reference: Stanford “Deep Generative Models”

Less Parameters: Bayesian Network

- Bayesian Network structure is a **Directed Acyclic Graph**, $G = (V, E)$

- Bayesian Network is given by (G, P),
  where P is a set of local **conditional probability distributions** for each node/vertex of G

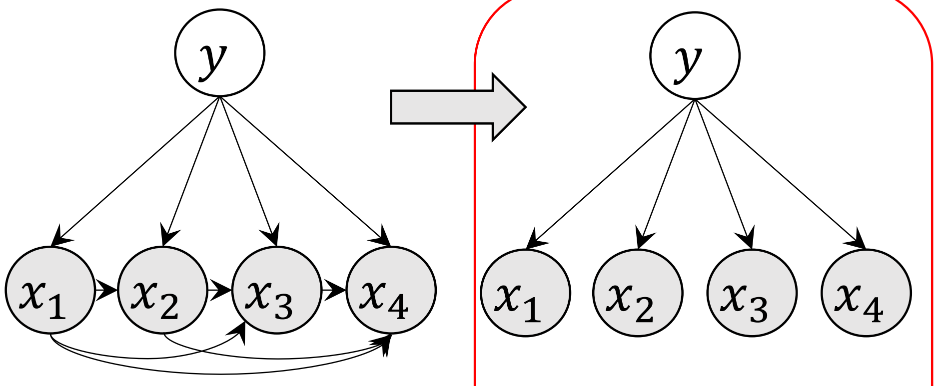
- Compute the P using data samples to "learn" the Bayesian Network

- Bayesian Network is also known as **Belief Network** and **Bayes Network**

- Problem of High-dimensional Data
- Less Parameters: Conditional Independence
- Less Parameters: Bayesian Network
- Naïve Bayes Classifier
- Discriminative vs. Generative Models
- Logistic Regression
- Deep Neural Networks
- Continuous Variables

Naïve Bayes Classifier

- How Bayesian Network performs inferencing? i.e., discriminative tasks?
- Support we have a binary classification problem, label $y = 0, 1$, features $\mathbf{x} = (x_1, x_2, x_3, x_4)$
- The probability distribution is $p(y, x_1, x_2, x_3, x_4)$
- Naïve Bayes Classifier assume that $\boxed{x_i \perp \mathbf{x}_{-i} | y}$, so that:



Generative Bayesian Network

Generative Naïve Bayes

Given Naïve Bayes Assumption:

$$\boxed{p(y, x_1, x_2, x_3, x_4) = p(y)\underbrace{p(x_1|\mathrm{y})p(x_2|\mathrm{y})\, p(x_3|\mathrm{y})p(x_4|\mathrm{y})}_{p(\mathbf{x}|y)}}$$

we need $p(y|\mathbf{x})$

Naïve Bayes Classifier

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}, y)}{p(\mathbf{x})} = \frac{p(y)p(\mathbf{x}|y)}{p(\mathbf{x})} \boxed{\propto p(y)p(x|y)}$$

$$\hat{y} = arg\max_y p(y|\mathbf{x}) = arg\max_y \frac{p(\mathbf{x},y)}{p(\mathbf{x})} = arg\max_y p(y)p(\mathbf{x}|y)$$

Given Naïve Bayes Assumption:

$$p(\mathbf{x}|y) = p(x_1|\mathrm{y})p(x_2|\mathrm{y})\, p(x_3|\mathrm{y})$$

Naïve Bayes Classifier

- Given $p(\mathbf{x}|y) = p(x_1|\mathrm{y})p(x_2|\mathrm{y})\, p(x_3|\mathrm{y})$, how to compute $p(Y|X)$?
- First, we can **estimate** the parameters from the training set:

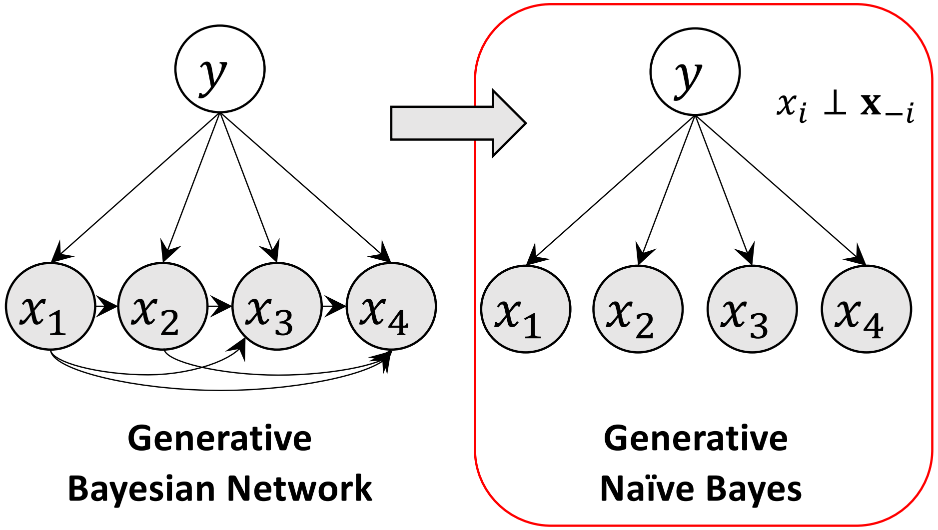
	$x_1 = 0$	$x_1 = 1$	$x_2 = 0$	$x_2 = 1$	$x_3 = 0$	$x_3 = 1$	$x_4 = 0$	$x_4 = 1$
$y = 0$	3	5	5	2	0	8	7	4
$y = 1$	1	0	3	10	7	4	2	5

  - $p(Y = 0) = \frac{3+5+5+2+8+7+4}{(3+5+5+2+8+7+4)+(1+3+10+7+4+2+5)}$
  - $p(x_1 = 0|Y = 0) = \frac{3}{3+5+5+2+8+7+4}$
  - ….
- Second, **predict** the probability of a label given an input with **Bayes rule**:
  - $p(Y = 0|x_1, x_2, x_3, x_4) = \frac{p(Y=0)\prod_{i=1}^{4} p(x_i|Y=0)}{\sum_{y=\{0,1\}} p(Y=y)\prod_{i=1}^{4} p(x_i|Y=y)}$

Naïve Bayes Classifier

- Limitation

Are the independence assumptions reasonable ??



- Problem of High-dimensional Data
- Less Parameters: Conditional Independence
- Less Parameters: Bayesian Network
- Naïve Bayes Classifier
- Discriminative vs. Generative Models
- Logistic Regression
- Deep Neural Networks
- Continuous Variables

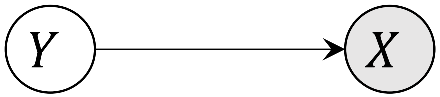
Discriminative vs. Generative Models

symmetry property $p(X, Y) = p(Y, X)$

- Given $p(Y, X) = p(X|Y)p(Y) = p(Y|X)p(X)$

- Discriminative: $X \rightarrow Y$, we only need to estimate the conditional distribution $P(Y|X)$
  without learning to model $P(X)$
  simply input X then output Y

- Generative: $Y \rightarrow X$, we need both $P(Y)$ and $P(X|Y)$ to compute $p(Y|X)$ via Bayes
  (see the Naïve Bayes Classifier as an example)



Discriminative vs. Generative Models

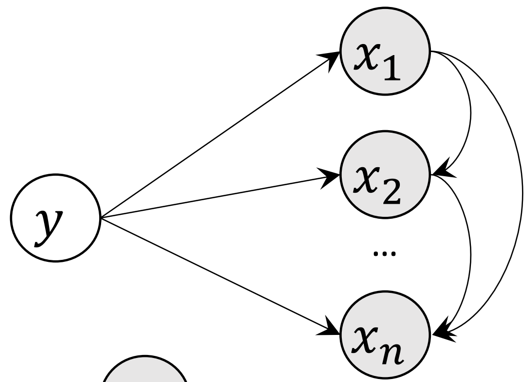
- Given a random vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$, the product rules can give us:

$$p(y, \mathbf{x}) = p(y)p(x_1|y)p(x_2|y, x_1) \dots p(x_n|y, x_1, x_2,\dots, x_{n-1})$$
$$p(y, \mathbf{x}) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2) \dots p(y|x_1, x_2,\dots, x_{n-1})$$

generative

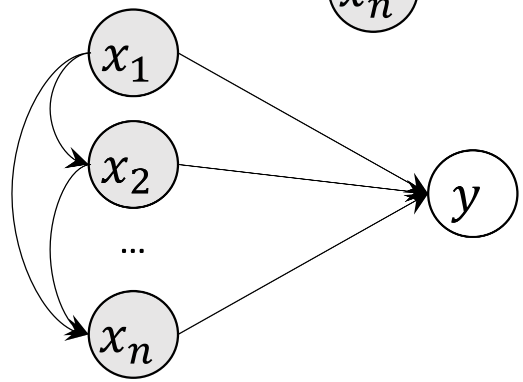
$p(y)$ is simple to estimate

but how to parametrise $p(x_i|y, x_1,\dots,x_{i-1})$?



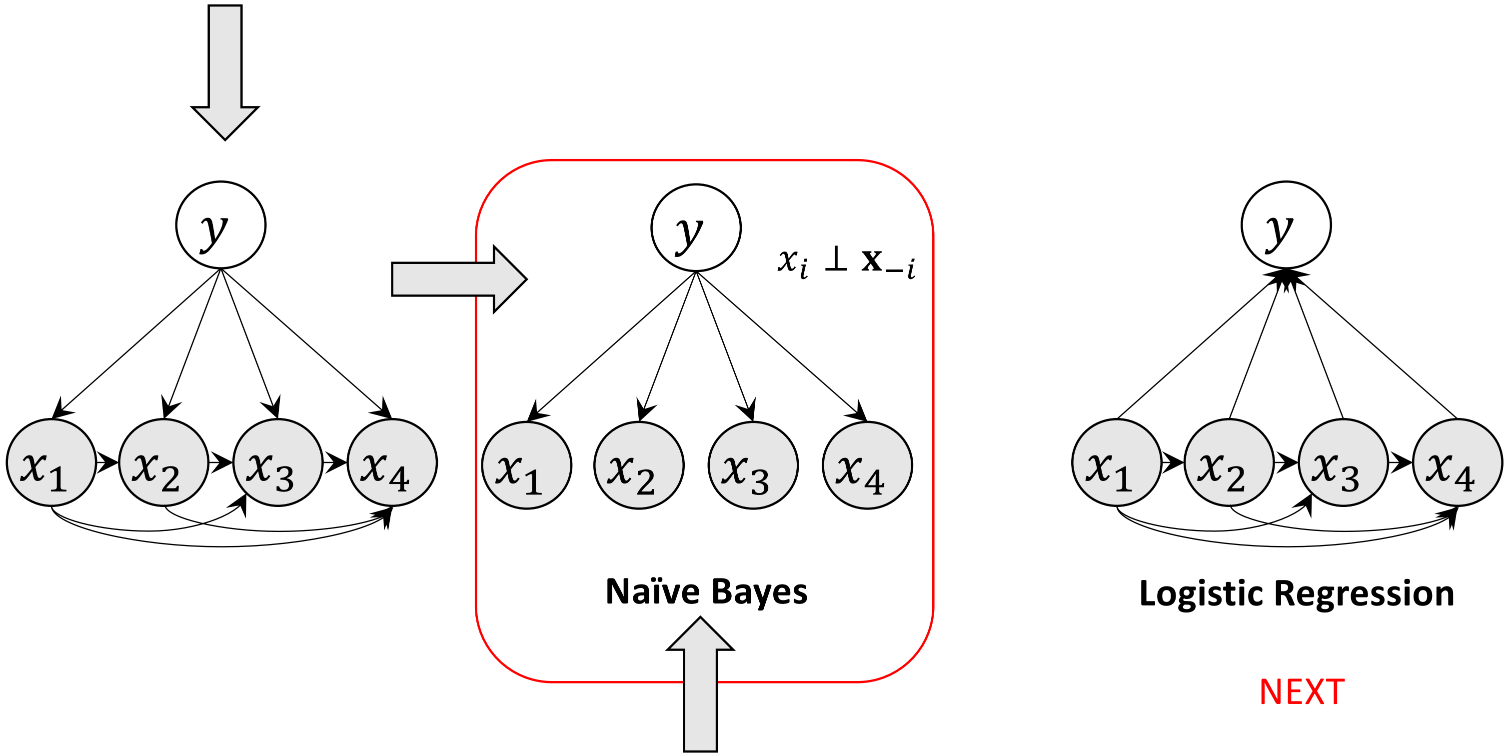
discriminative

only need to parametrise $p(y|x_1,\dots,x_{n-1})$



Discriminative vs. Generative Models

parametrise $p(x_i|y, x_1,\ldots,x_{i-1})$ without independent assumptions



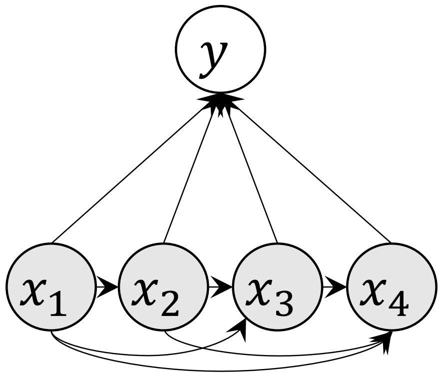
parametrise $p(x_i|y, x_1,\ldots,x_{i-1})$ with independent assumptions

- Problem of High-dimensional Data
- Less Parameters: Conditional Independence
- Less Parameters: Bayesian Network
- Naïve Bayes Classifier
- Discriminative vs. Generative Models
- Logistic Regression
- Deep Neural Networks
- Continuous Variables

Logistic Regression

- Parameterise the $p(Y|X)$ without independence assumptions

only need to parametrise $p(y|x_1,\ldots,x_{n-1})$



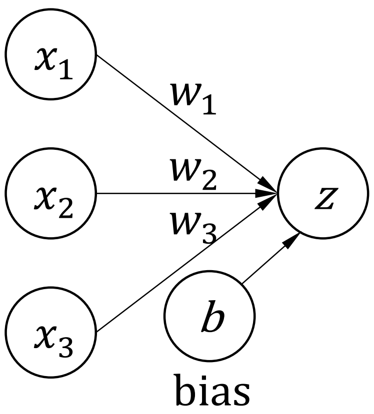
Logistic Regression

Logistic Regression

- (only need to) parameterise the $p(Y|X)$ without independence assumptions

$$\boxed{p(Y = 1|\mathbf{x}, \mathbf{w}, b) = f(\mathbf{x}, \mathbf{w}, b)}$$

input layer *output layer*



$$z = x_1w_1 + x_2w_2 + x_3w_3 + b$$

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

$$z = \boldsymbol{w}^T\boldsymbol{x} + b$$

$$z = [w_1 \quad w_2 \quad w_3] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + b$$

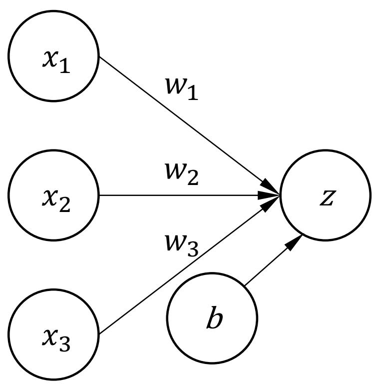
Logistic Regression

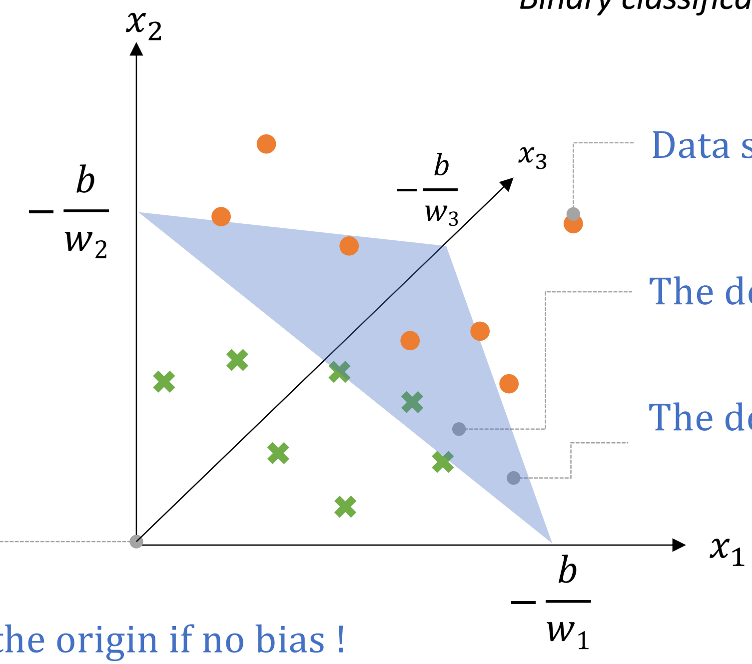
- (only need to) parameterise the $p(Y|X)$ without independence assumptions

$$z = x_1 w_1 + x_2 w_2 + x_3 w_3 + b$$

Binary classification: $y = \begin{cases} 0, if\ z \le 0 \\ 1, if\ z > 0 \end{cases}$

input layer *output layer*





Data samples with three features (x_1, x_2, x_3)

The decision boundary is a surface for $z = 0$

The decision boundary can be shifted left or right via the bias

The decision boundary must cross the origin if no bias !

Logistic Regression

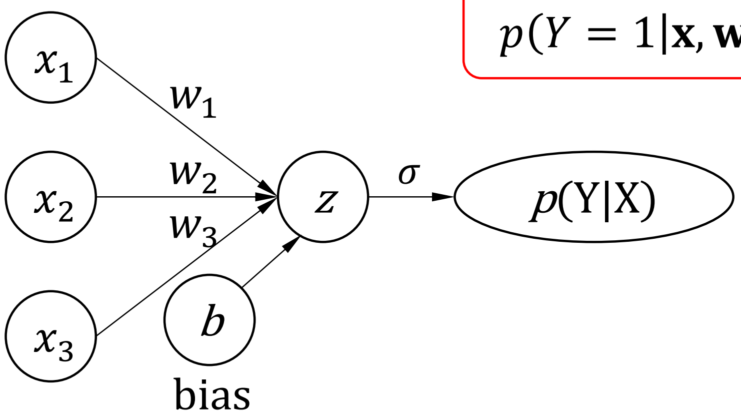
- (only need to) parameterise the $p(Y|X)$ without independence assumptions

$$\boxed{p(Y=1|\mathbf{x},\mathbf{w},b)=f(\mathbf{x},\mathbf{w},b)}$$

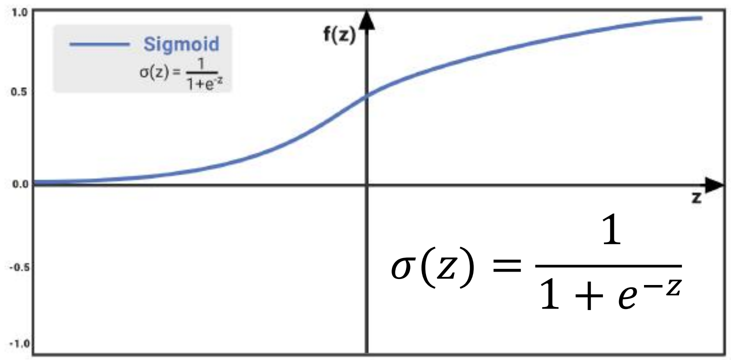
$$z=\mathbf{w}^T\mathbf{x}+b$$

input layer *output layer*

$$\boxed{p(Y=1|\mathbf{x},\mathbf{w},b)=\sigma(\mathbf{w}^T\mathbf{x}+b)}$$



Sigmoid/Logistic function



Logistic Regression

- Logistic regression does not require independence assumptions $x_i \perp \mathbf{x}_{-i}$, like Naïve Bayes
- Example, in spam classification, $x_1 = 1$[“bank” exists] and $x_2 = 1$[“account” exists]

If “bank” and “account” always appear together,

Naïve Bayes will **count this evidence twice**:

$$p(y, x_1, x_2, x_3) = p(y)p(x_1|\mathrm{y})p(x_2|\mathrm{y})\, p(x_3|\mathrm{y})$$

$$p(x_1|y) = p(x_2|y)$$

Logistic regressive can set either w_1 or w_2 to **zero** to ignore one of it!!

Reference: Stanford “Deep Generative Models”

Logistic Regression

- Discriminative model is powerful, so what is the advantage of generative model?
  - Discriminative models $p(Y|X)$ require all X are observed,
    fail to work if some inputs are missing!
  - Generative models $p(Y|X) = \frac{p(Y,X)}{p(X)} = \frac{p(Y)p(X|Y)}{p(X)} \propto p(Y)p(X|Y)$
    when some input are unobserved, still allow us to compute $p(Y|X)$
    e.g., Naive Bayes

	$x_1 = 0$	$x_1 = 1$	$x_2 = 0$	$x_2 = 1$	$x_3 = 0$	$x_3 = 1$	$x_4 = 0$	$x_4 = 1$
$y = 0$	3	5	5	2	0	~~8~~	~~7~~	4
$y = 1$	1	0	3	10	7	~~4~~	~~2~~	5

- $p(Y = 0) = \frac{3+5+5+2+\cancel{8+7}+4}{(3+5+5+2+\cancel{8+7}+4)+(1+3+10+7+\cancel{4+2}+5)}$
- $p(x_1 = 0|Y = 0) = \frac{3}{3+5+5+2+\cancel{8+7}+4}$

- Problem of High-dimensional Data
- Less Parameters: Conditional Independence
- Less Parameters: Bayesian Network
- Naïve Bayes Classifier
- Discriminative vs. Generative Models
- Logistic Regression
- Deep Neural Networks
- Continuous Variables

Deep Neural Network

- Logistic regression parameterises the $p(Y|X)$ without independence assumptions

$$p(Y = 1|\mathbf{x}, \mathbf{w}, b) = f(\mathbf{x}, \mathbf{w}, b)$$

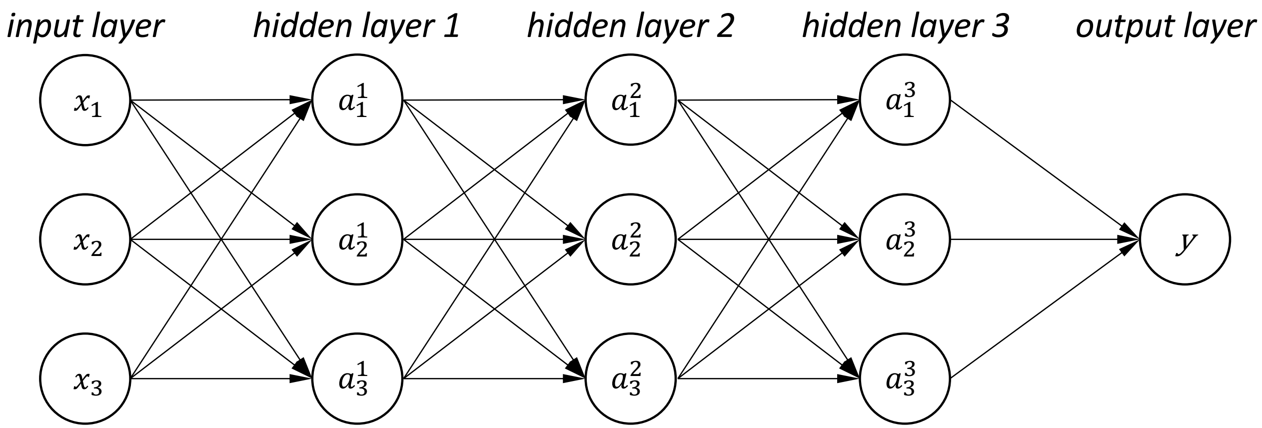
but logistic regression is a **linear dependence** (between input and output) which might be too simple

Non-linear dependence is better …

$$p_{Neural}(Y = 1|\mathbf{x}, \theta) = f(\mathbf{x}, \theta)$$

Deep Neural Network

More parameters and layers, better representation capacity …



More powerful than logistic regression

Deep Neural Network

- Naïve Bayes

$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)p(x_4|x_1, x_2, x_3)$$
$$\approx p(x_1)p(x_2|x_1)p(x_3|\cancel{x_1}, x_2)p(x_4|\cancel{x_1, x_2}, x_3)$$
$$\approx p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_3)$$

- Deep Neural Network

$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)p(x_4|x_1, x_2, x_3)$$
$$\approx p(x_1)p(x_2|x_1)p_{Neural}(x_3|x_1, x_2)pNeu_{ral}(x_4|x_1, x_2, x_3)$$

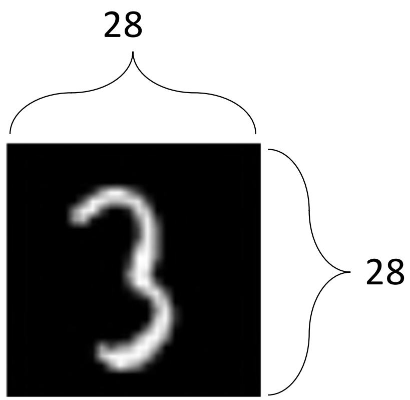
- Problem of High-dimensional Data
- Less Parameters: Conditional Independence
- Less Parameters: Bayesian Network
- Naïve Bayes Classifier
- Discriminative vs. Generative Models
- Logistic Regression
- Deep Neural Networks
- Continuous Variables

Continuous Variables

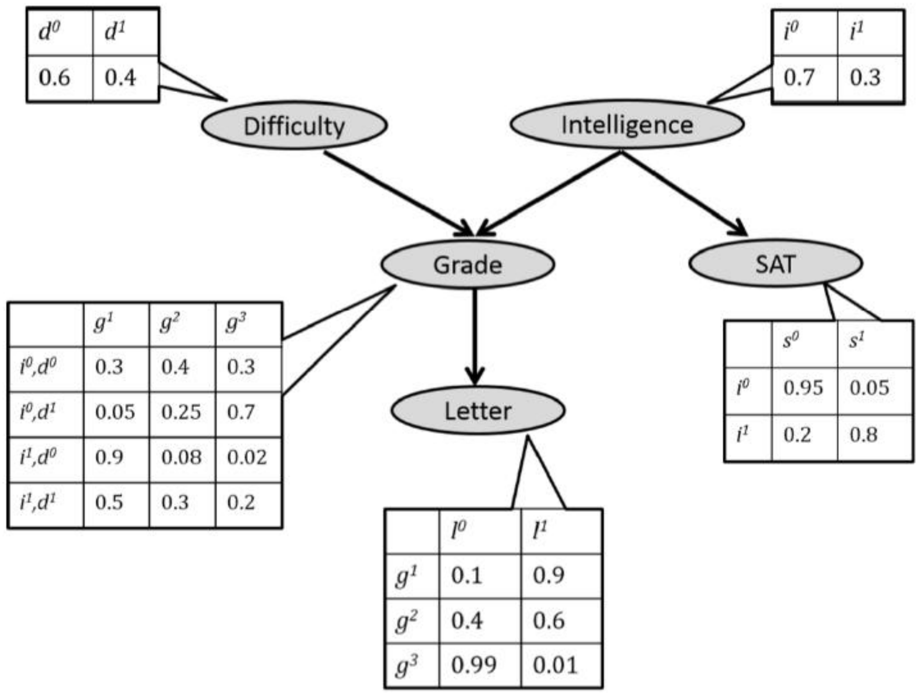
- **Discrete Variables**

The below examples both use discrete variables, but there are many variables are continuous!

e.g., age, height …



784 random binary variables



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

Continuous Variables

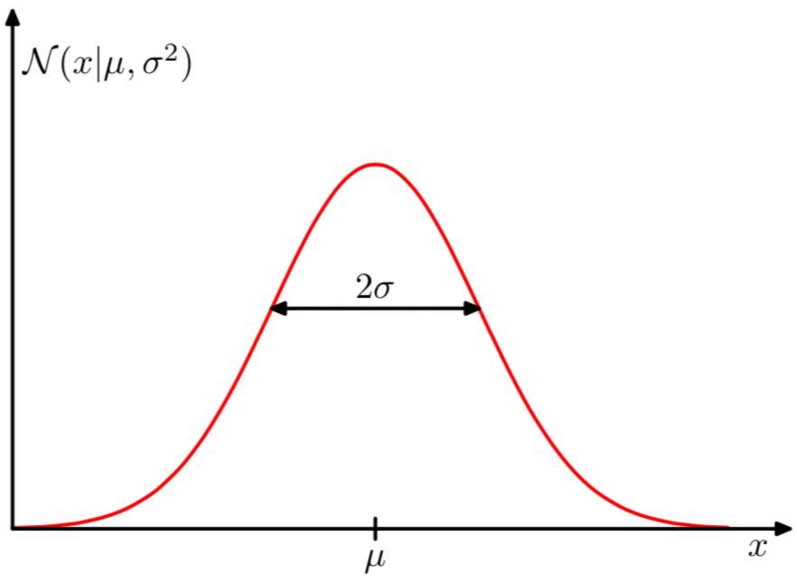
- **Represent Continuous Variables**

If x is a continuous variable, we can represent it with its **probability density function (PDF)** instead of a **table** anymore ..

Continuous Variables

- **Represent Continuous Variables**

Consider x is a random **float-point** variable to represent “age”, we can use 1-D Gaussian to parameterised the density.



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

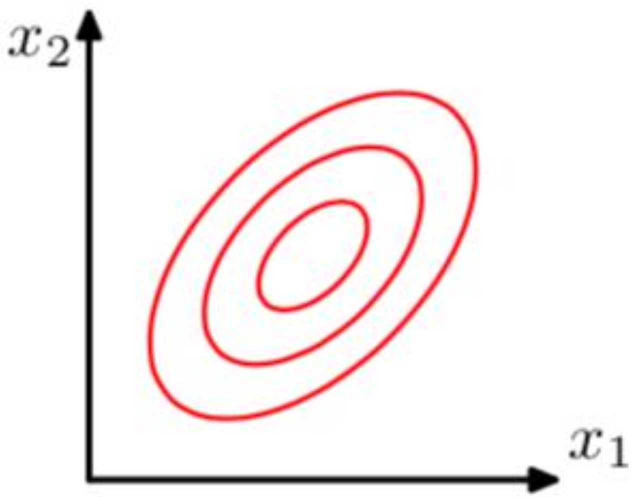
$$\mathcal{N}(x|\mu,\sigma^2) > 0$$

μ: mean
σ: standard deviation
σ^2: variance
$\beta = \frac{1}{\sigma^2}$: precision

Continuous Variables

- **Represent Continuous Variables**

Consider $\mathbf{x}$ is a random **float-point** vector to represent “age”, “height”, “weight” …..
it can be a **joint probability density function**
we can use **D-dimensional Gaussian** to parameterize it
(a.k.a Multivariable Gaussian)



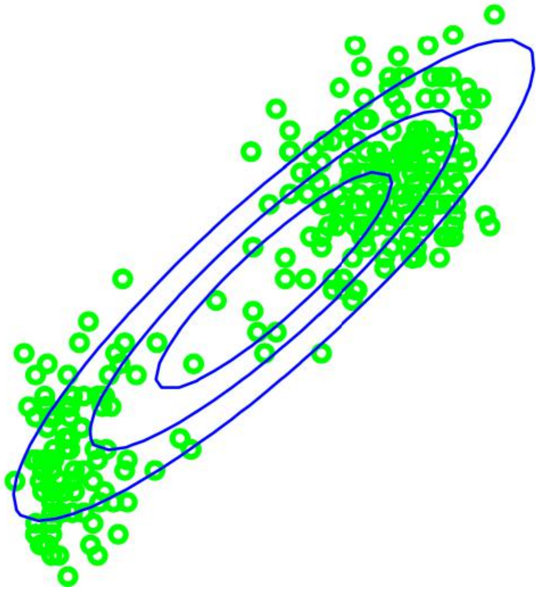
$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

μ is called the mean
Σ is called the covariance
$|\Sigma|$ denotes the determinant of Σ

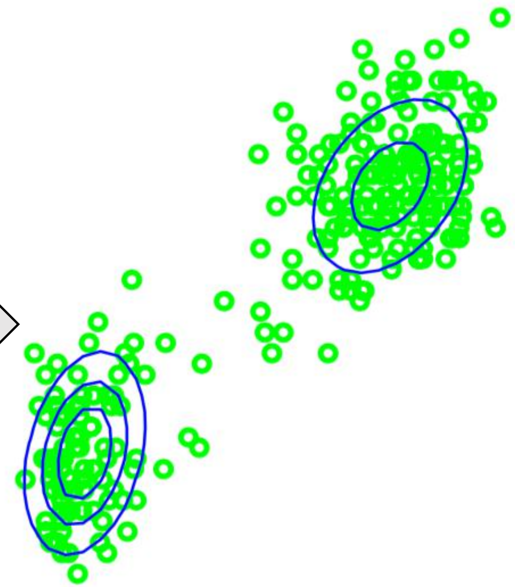
Continuous Variables

- **Represent Continuous Variables**

Consider $\mathbf{x}$ is a random **float-point** vector to represent "age", "height", "weight" …..



Sometime, **mixture Gaussian** is better



Data Representation

How to do representation

- Problem of High-dimensional Data
- Less Parameters: Conditional Independence
- Less Parameters: Bayesian Network

How to do inference

- Naïve Bayes Classifier
- Discriminative vs. Generative Models
- Logistic Regression

How to be better

- Deep Neural Networks
- Continuous Variables

Thanks