

## Anomaly detection example

→ Fraud detection:

→  $x^{(i)}$  = features of user  $i$ 's activities

→ Model  $p(x)$  from data.

→ Identify unusual users by checking which have  $p(x) < \varepsilon$

$$\begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{matrix} \quad p(x)$$

## Anomaly detection example

→ Fraud detection:

→  $x^{(i)}$  = features of user  $i$ 's activities

→ Model  $p(x)$  from data.

→ Identify unusual users by checking which have  $\underline{p(x) < \varepsilon}$

→ Manufacturing

→ Monitoring computers in a data center.

→  $x^{(i)}$  = features of machine  $i$

$x_1$  = memory use,  $x_2$  = number of disk accesses/sec,

$x_3$  = CPU load,  $x_4$  = CPU load/network traffic.

...

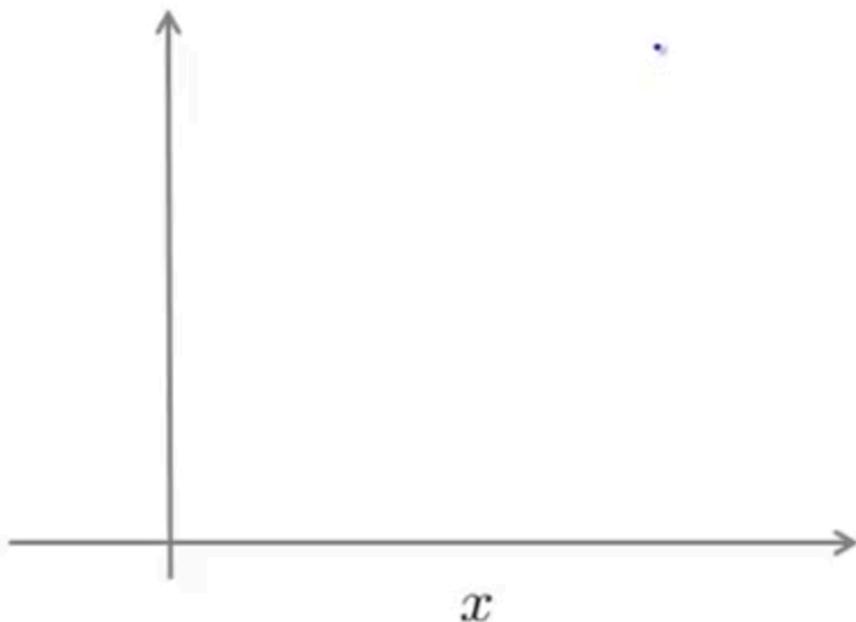
$\underline{p(x) < \varepsilon}$

$x_1$   
 $x_2$   
 $x_3$   
 $x_4$

$p(x)$

## Gaussian (Normal) distribution

Say  $x \in \mathbb{R}$ . If  $x$  is distributed Gaussian with mean  $\mu$ , variance  $\sigma^2$ .

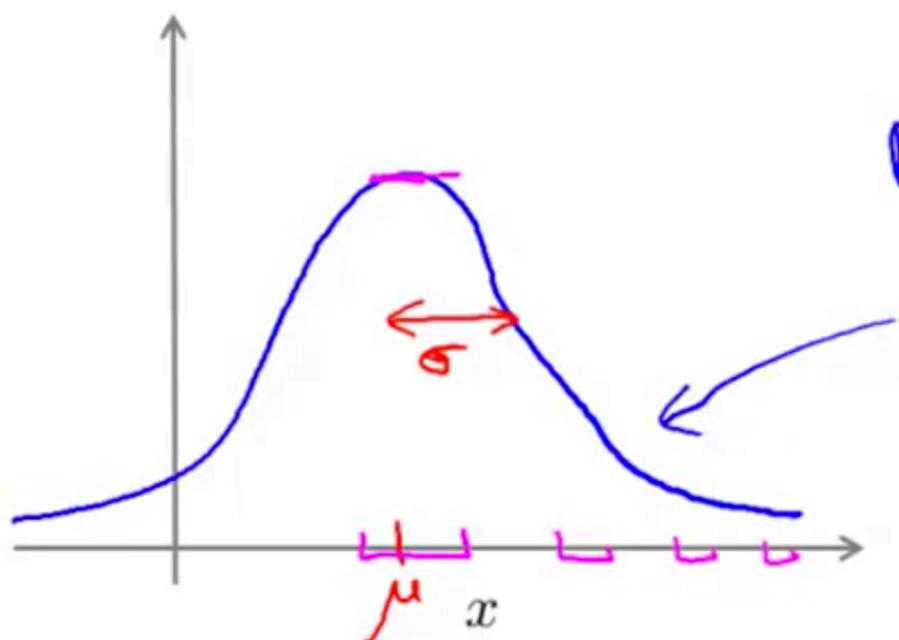


## Gaussian (Normal) distribution

Say  $\underline{x} \in \mathbb{R}$ . If  $x$  is distributed Gaussian with mean  $\mu$ , variance  $\sigma^2$ .

$$x \sim \mathcal{N}(\mu, \sigma^2)$$

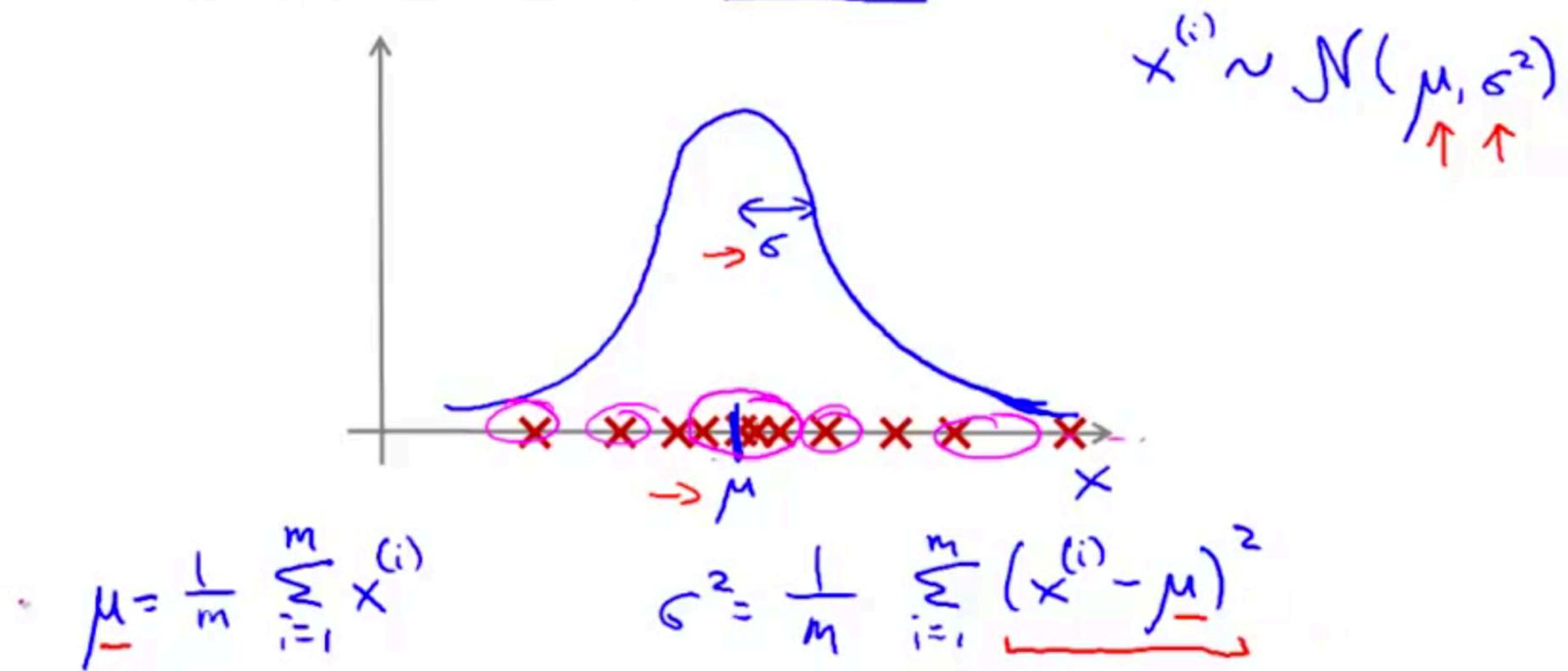
↖ "distributed as"



$$\begin{aligned} p(x; \mu, \sigma^2) \\ = \frac{1}{\sqrt{2\pi} \sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \end{aligned}$$

## Parameter estimation

→ Dataset:  $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$   $x^{(i)} \in \mathbb{R}$



## Density estimation

→ Training set:  $\{x^{(1)}, \dots, x^{(m)}\}$

Each example is  $x \in \mathbb{R}^n$

$$x_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$$

$$x_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$$

$$x_3 \sim \mathcal{N}(\mu_3, \sigma_3^2)$$

$p(x)$

$$= \boxed{p(x_1; \mu_1, \sigma_1^2) p(x_2; \mu_2, \sigma_2^2) p(x_3; \mu_3, \sigma_3^2) \cdots p(x_n; \mu_n, \sigma_n^2)}$$

$$= \boxed{\prod_{j=1}^n p(x_j; \mu_j, \sigma_j^2)}$$

$$\sum_{i=1}^n i = 1+2+3+\dots+n$$

$$\prod_{i=1}^n i = 1 \times 2 \times 3 \times \dots \times n$$

Given a training set  $\{x^{(1)}, \dots, x^{(m)}\}$ , how would you estimate each  $\mu_j$  and  $\sigma_j^2$  (Note  $\mu_j \in \mathbb{R}, \sigma_j^2 \in \mathbb{R}.$ )

- $\mu_j = \frac{1}{m} \sum_{i=1}^m x^{(i)}, \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu)^2$
- $\mu_j = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)})^2, \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)} - \mu_j)^2$
- $\mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)}, \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu)^2$
- $\mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)}, \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)} - \mu_j)^2$

**Well done!**

Continue

## Anomaly detection algorithm

→ 1. Choose features  $x_i$  that you think might be indicative of anomalous examples.

$$\{x^{(1)}, \dots, x^{(m)}\}$$

→ 2. Fit parameters  $\mu_1, \dots, \mu_n, \sigma_1^2, \dots, \sigma_n^2$

$$\rightarrow \mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)}$$

$$\rightarrow \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)} - \mu_j)^2$$

3. Given new example  $x$ , compute  $p(x)$ :

$$p(x) = \prod_{j=1}^n p(x_j; \mu_j, \sigma_j^2) = \prod_{j=1}^n \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}\right)$$

Anomaly if  $p(x) < \varepsilon$

## Anomaly detection algorithm

- 1. Choose features  $x_i$  that you think might be indicative of anomalous examples.  $\{x^{(1)}, \dots, x^{(m)}\}$

- 2. Fit parameters  $\mu_1, \dots, \mu_n, \sigma_1^2, \dots, \sigma_n^2$

$$\rightarrow \boxed{\mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)}}$$

$$\rightarrow \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)} - \mu_j)^2$$

$$p(x_j; \mu_j, \sigma_j^2)$$

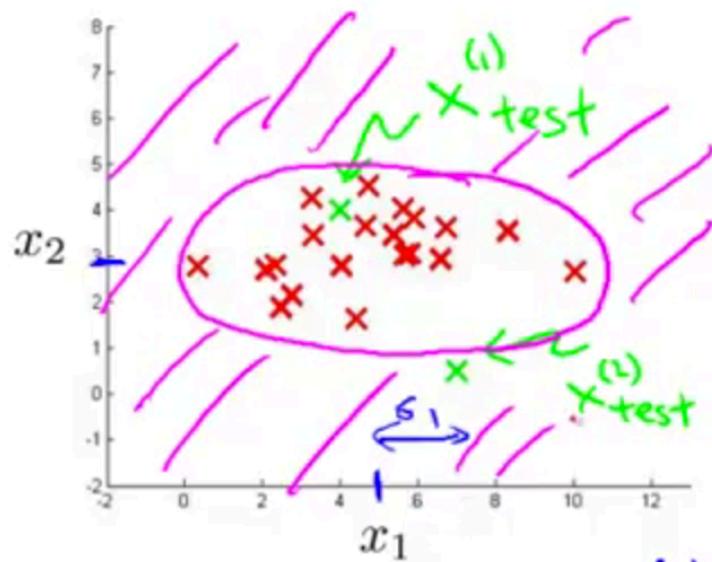
$$\mu_1, \mu_2, \dots, \mu_n$$
$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix} = \frac{1}{m} \sum_{i=1}^m x^{(i)}$$

- 3. Given new example  $x$ , compute  $p(x)$ :

$$p(x) = \prod_{j=1}^n p(x_j; \mu_j, \sigma_j^2) = \prod_{j=1}^n \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}\right)$$

Anomaly if  $\underline{p(x) < \varepsilon}$

## Anomaly detection example



$$\begin{aligned} \mu_1 &= 5, \sigma_1 = 2 \\ \mu_2 &= 3, \sigma_2 = 1 \end{aligned}$$

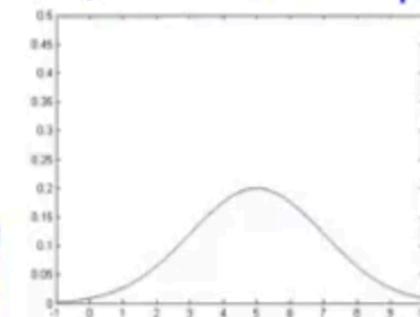
$\sigma_1^2, \sigma_2^2 = 4$

$$\rightarrow p(x) = p(x_1; \mu_1, \sigma_1^2) \times p(x_2; \mu_2, \sigma_2^2)$$

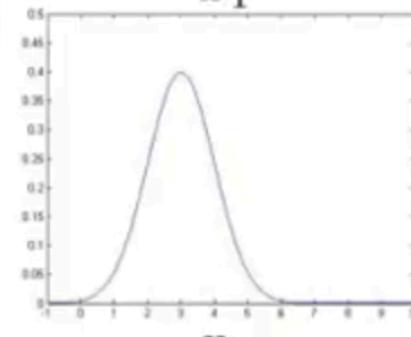
$$p(x_1; \mu_1, \sigma_1^2)$$



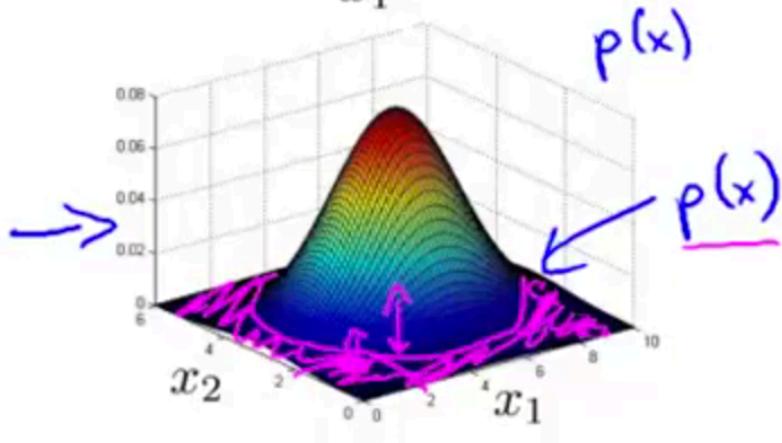
$$p(x_2; \mu_2, \sigma_2^2)$$



$x_1$



$x_2$



$$\varepsilon = 0.02$$

$$p(x_{test}^{(1)}) = 0.0426 \geq \varepsilon$$

$$p(x_{test}^{(2)}) = 0.0021 < \varepsilon$$

## Aircraft engines motivating example

- 10000 good (normal) engines
- 20 flawed engines (anomalous)  $\frac{2-50}{y=1}$   
 $\mu_1, \sigma^2_1, \dots, \mu_n, \sigma^2_n$
- Training set: 6000 good engines ( $y = 0$ )  $p(x) = p(x_1; \mu_1, \sigma^2_1) \dots p(x_n; \mu_n, \sigma^2_n)$   
CV: 2000 good engines ( $y = 0$ ), 10 anomalous ( $y = 1$ )  
Test: 2000 good engines ( $y = 0$ ), 10 anomalous ( $y = 1$ )

## Algorithm evaluation

- Fit model  $p(x)$  on training set  $\{x^{(1)}, \dots, x^{(m)}\}$
- On a cross validation/test example  $x$ , predict

$$y = \begin{cases} 1 & \text{if } p(x) < \varepsilon \text{ (anomaly)} \\ 0 & \text{if } p(x) \geq \varepsilon \text{ (normal)} \end{cases}$$

Possible evaluation metrics:

- True positive, false positive, false negative, true negative
- Precision/Recall
- $F_1$ -score

Can also use cross validation set to choose parameter  $\varepsilon$

## Algorithm evaluation

- Fit model  $p(x)$  on training set  $\{x^{(1)}, \dots, x^{(m)}\}$
- On a cross validation/test example  $x$ , predict

$(x_{\text{test}}^{(i)}, y_{\text{test}}^{(i)})$

$$y = \begin{cases} 1 & \text{if } p(x) < \varepsilon \text{ (anomaly)} \\ 0 & \text{if } p(x) \geq \varepsilon \text{ (normal)} \end{cases}$$

$y=0$

Possible evaluation metrics:

- - True positive, false positive, false negative, true negative
- - Precision/Recall
- -  $F_1$ -score

Can also use cross validation set to choose parameter  $\underline{\varepsilon}$

## Anomaly detection

- Very small number of positive examples ( $y = 1$ ). (0-20 is common).
- Large number of negative ( $y = 0$ ) examples.  $p(x)$
- Many different “types” of anomalies. Hard for any algorithm to learn from positive examples what the anomalies look like;
- future anomalies may look nothing like any of the anomalous examples we've seen so far.

vs.

## Supervised learning

- Large number of positive and negative examples.
- Enough positive examples for algorithm to get a sense of what positive examples are like, future positive examples likely to be similar to ones in training set.

## Anomaly detection

- • Fraud detection       $y=1$
  - • Manufacturing (e.g. aircraft engines)
  - • Monitoring machines in a data center
- ⋮

vs.

## Supervised learning

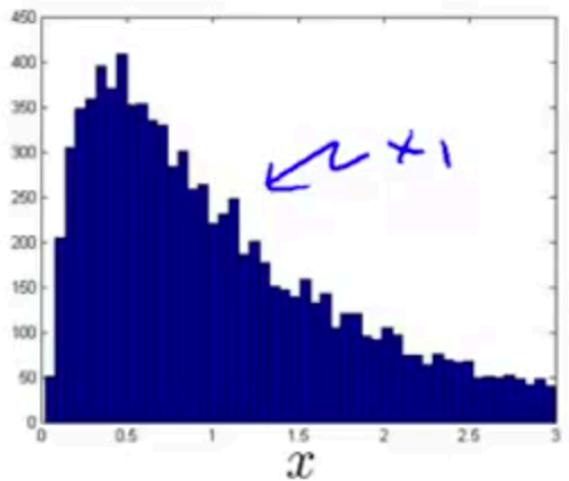
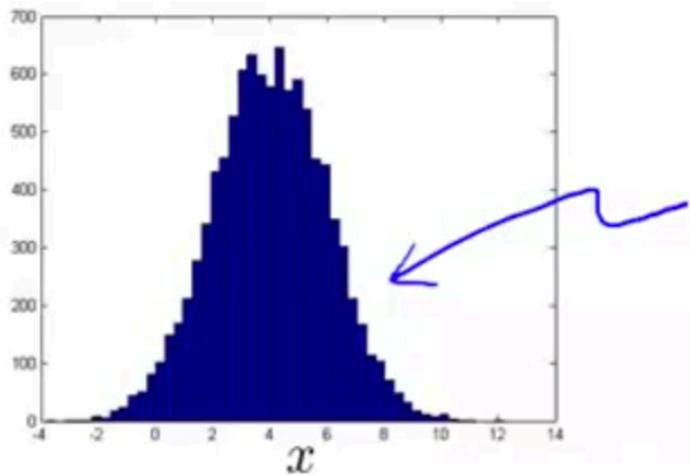
- Email spam classification ←
  - Weather prediction ←  
(sunny/rainy/etc).
  - Cancer classification ←
- ⋮



6:57 / 7:36



## Non-gaussian features



$\log(x)$

$$p(x_i; \underline{\mu}_i, \underline{\sigma}^2_i)$$

hist

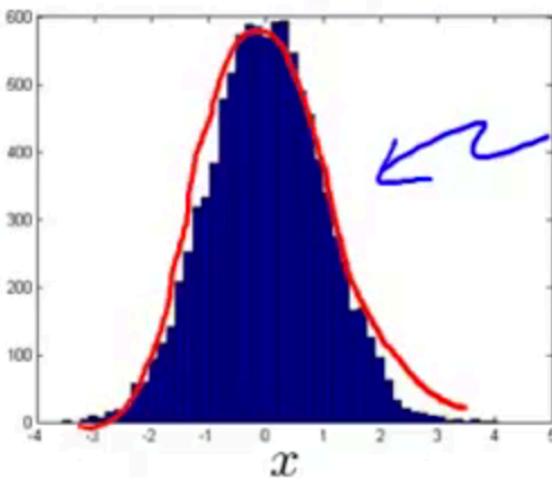
$$x_1 \leftarrow \log(x_i)$$

$$x_2 \leftarrow \log(x_2 + 1)$$

$$x_3 \leftarrow \sqrt{x_3} = x_3^{\frac{1}{2}}$$

$$x_4 \leftarrow x_4^{\frac{1}{3}} =$$

$$\log(x_2 + 1) \downarrow$$

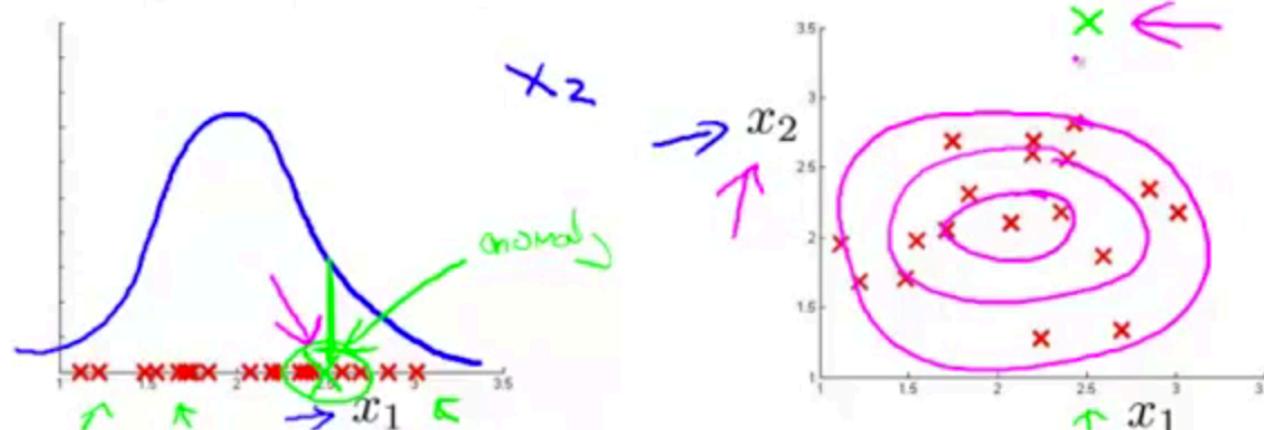


## → Error analysis for anomaly detection

- Want  $p(x)$  large for normal examples  $x$ .
- $p(x)$  small for anomalous examples  $x$ .

Most common problem:

- $p(x)$  is comparable (say, both large) for normal and anomalous examples

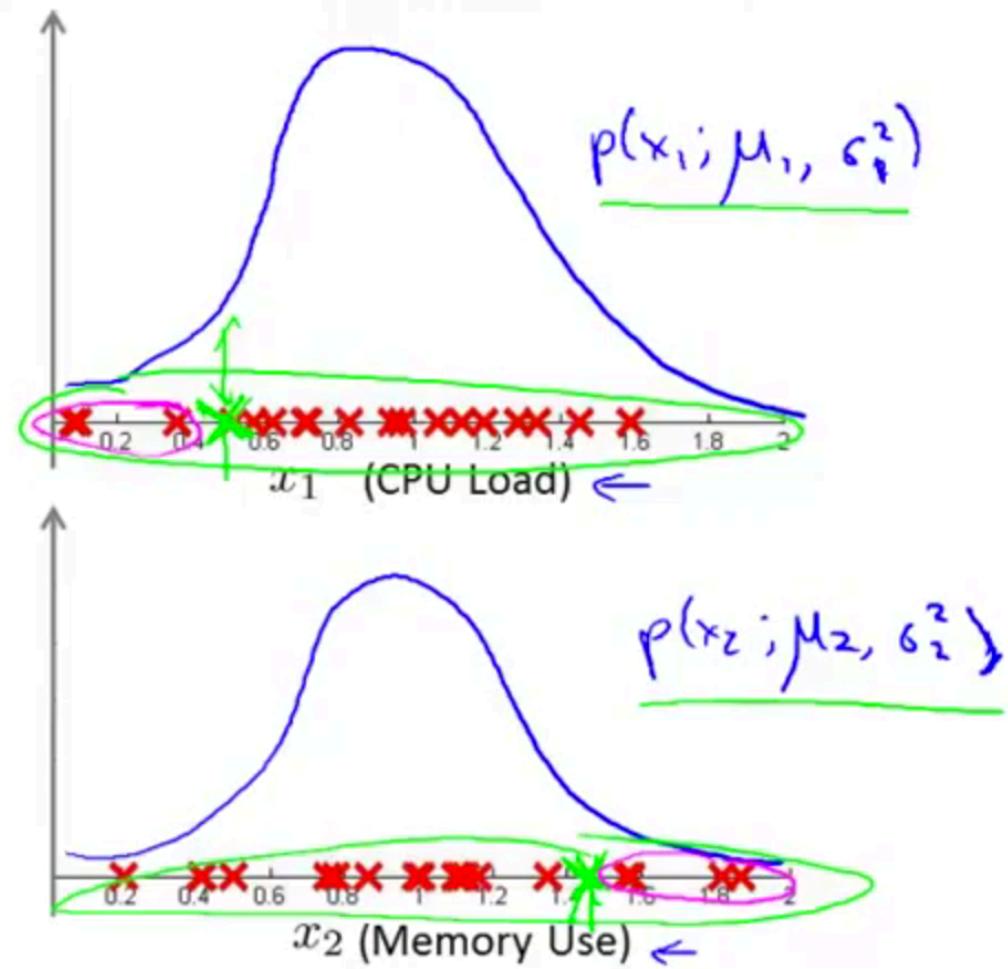
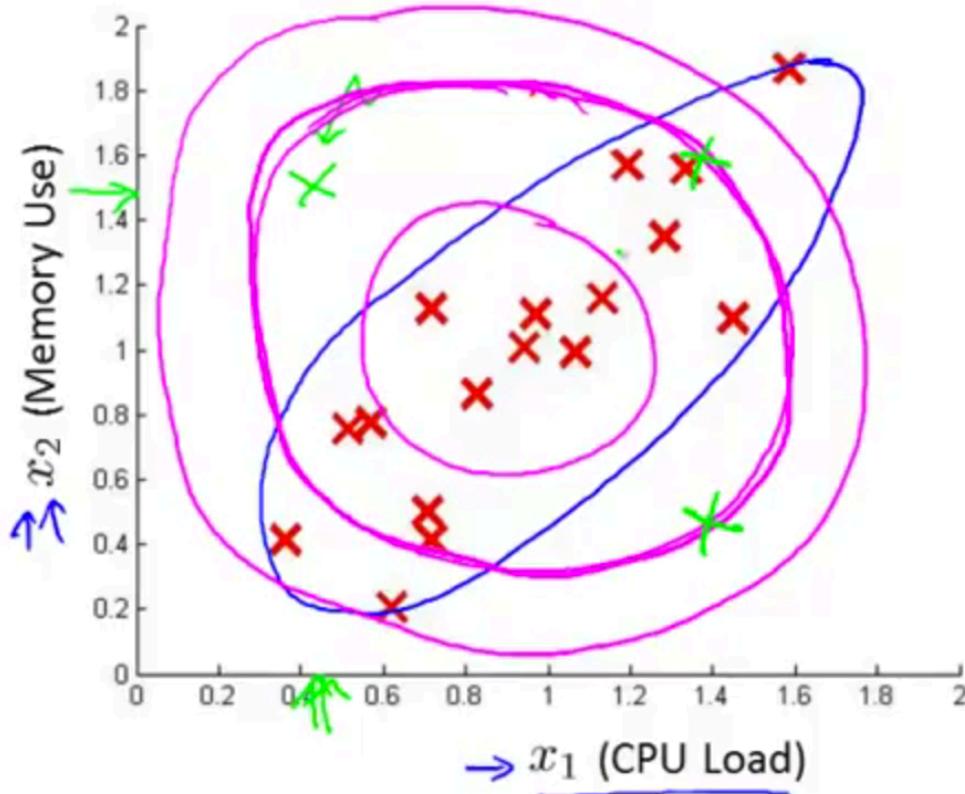


- Monitoring computers in a data center
- Choose features that might take on unusually large or small values in the event of an anomaly.
  - $x_1$  = memory use of computer
  - $x_2$  = number of disk accesses/sec
  - $x_3$  = CPU load ←
  - $x_4$  = network traffic ←

$$x_5 = \frac{\text{CPU load}}{\text{network traffic}}$$

$$x_6 = \frac{(\text{CPU load})^2}{\text{network traffic}}$$

## Motivating example: Monitoring machines in a data center



## Multivariate Gaussian (Normal) distribution

→  $x \in \mathbb{R}^n$ . Don't model  $p(x_1), p(x_2), \dots$ , etc. separately.  
Model  $p(x)$  all in one go.

Parameters:  $\mu \in \mathbb{R}^n$ ,  $\Sigma \in \mathbb{R}^{n \times n}$  (covariance matrix)

$$p(x; \mu, \Sigma) =$$

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

$$|\Sigma| = \text{determinant of } \Sigma \quad | \det(\Sigma)|$$

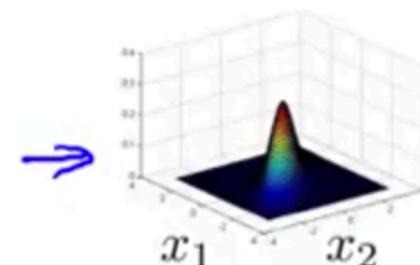
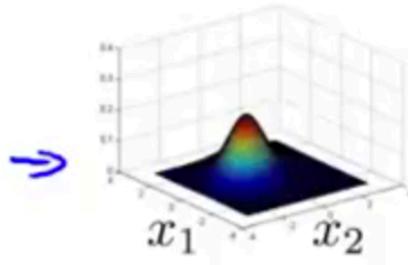
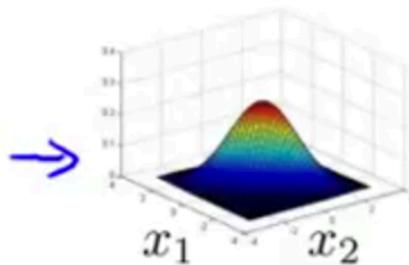
## Multivariate Gaussian (Normal) distribution

Parameters  $\underline{\mu, \Sigma}$

$$\mu \in \mathbb{R}^n$$

$$\Sigma \in \mathbb{R}^{n \times n}$$

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



Parameter fitting:

Given training set  $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\} \leftarrow x \in \mathbb{R}^n$

$$\rightarrow \boxed{\mu} = \frac{1}{m} \sum_{i=1}^m x^{(i)}$$
$$\rightarrow \boxed{\Sigma} = \frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu)(x^{(i)} - \mu)^T$$

## Anomaly detection with the multivariate Gaussian

1. Fit model  $p(x)$  by setting

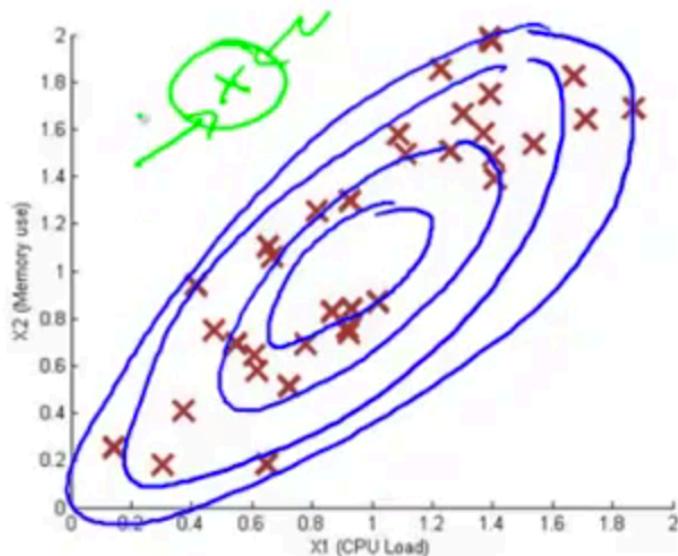
$$\mu = \frac{1}{m} \sum_{i=1}^m x^{(i)}$$

$$\Sigma = \frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu)(x^{(i)} - \mu)^T$$

2. Given a new example  $x$ , compute

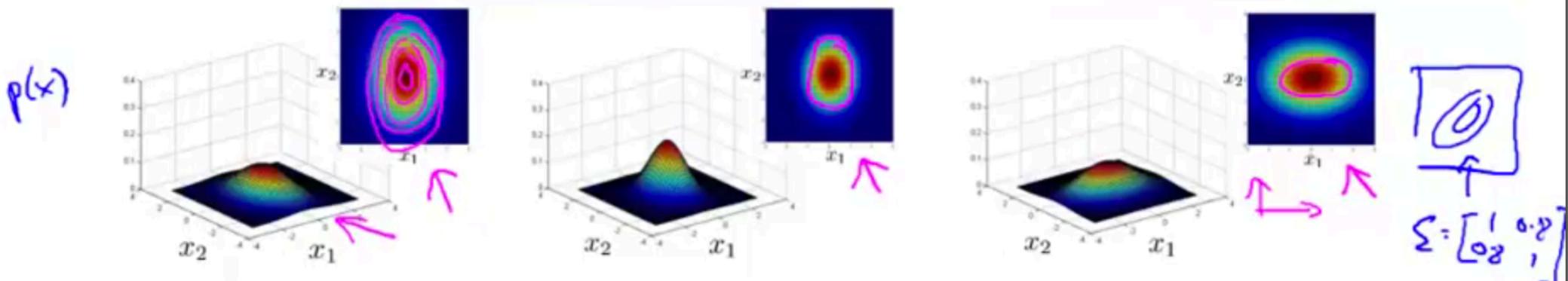
$$p(x) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right)$$

Flag an anomaly if  $p(x) < \varepsilon$



## Relationship to original model

Original model:  $p(x) = p(x_1; \mu_1, \sigma_1^2) \times p(x_2; \mu_2, \sigma_2^2) \times \cdots \times p(x_n; \mu_n, \sigma_n^2)$



Corresponds to multivariate Gaussian

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

where

$$\Sigma = \begin{bmatrix} \sigma_1^2 & & & \\ & \ddots & & \\ & & \sigma_n^2 & \\ & & & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

## → Original model

$$p(x_1; \mu_1, \sigma_1^2) \times \cdots \times p(x_n; \mu_n, \sigma_n^2)$$

Manually create features to capture anomalies where  $x_1, x_2$  take unusual combinations of values.

$$\rightarrow X_3 = \frac{x_1}{x_2} = \frac{\text{CPU load}}{\text{memory}}$$

→ Computationally cheaper (alternatively, scales better to large  $n$ )     $n=10,000$ ,     $n=100,000$

OK even if  $m$  (training set size) is small

## vs. → Multivariate Gaussian

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

→ Automatically captures correlations between features

$$\Sigma \in \mathbb{R}^{n \times n}$$

$$\Sigma^{-1}$$

Computationally more expensive

$$\Sigma \sim \frac{n^2}{2}$$

Must have  $m > n$  or else  $\Sigma$  is non-invertible.

$$m \geq 10n$$