

Machine Learning

# Anomaly detection

---

Problem  
motivation

## Anomaly detection example

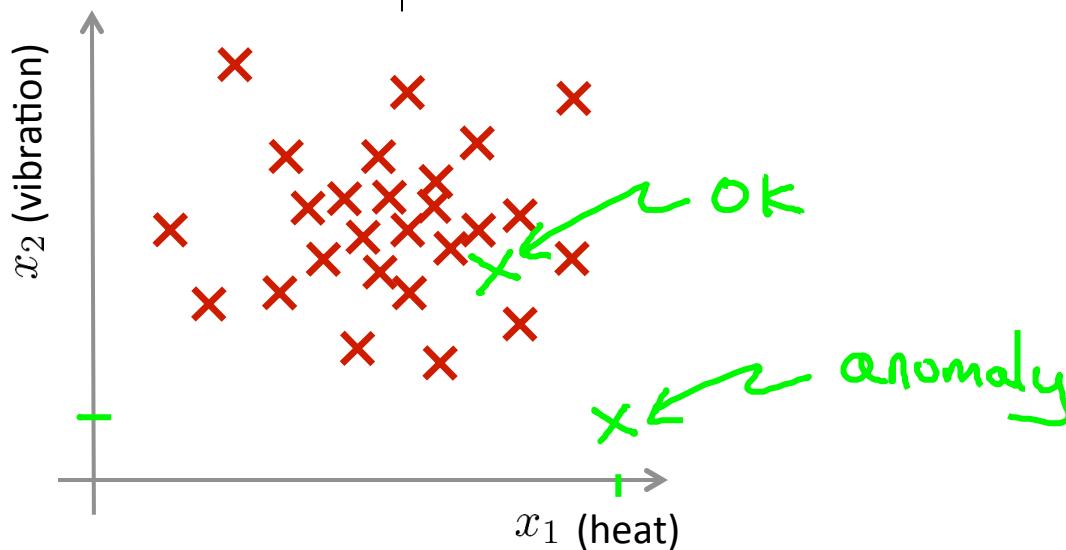
Aircraft engine features:

- $x_1$  = heat generated
- $x_2$  = vibration intensity

...

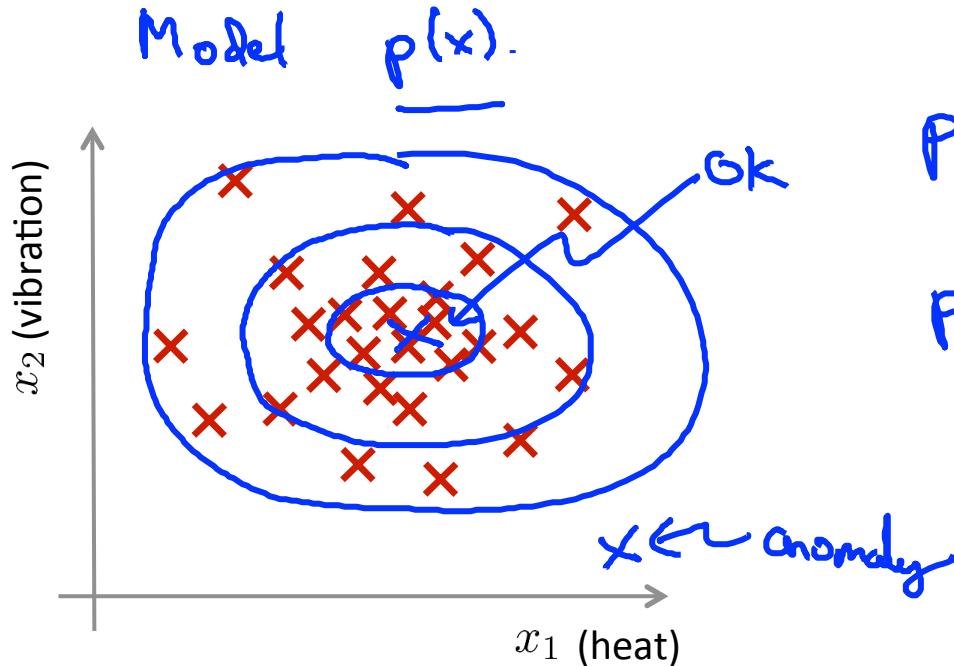
Dataset:  $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$

New engine:  $x_{test}$



# Density estimation

- Dataset:  $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$
- Is  $x_{test}$  anomalous?



$p(x_{test}) < \varepsilon \rightarrow \text{flag anomaly}$

$p(x_{test}) \geq \varepsilon \rightarrow \text{OK}$

## 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  $p(x) < \varepsilon$

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

→ Manufacturing

→ Monitoring computers in a data center.

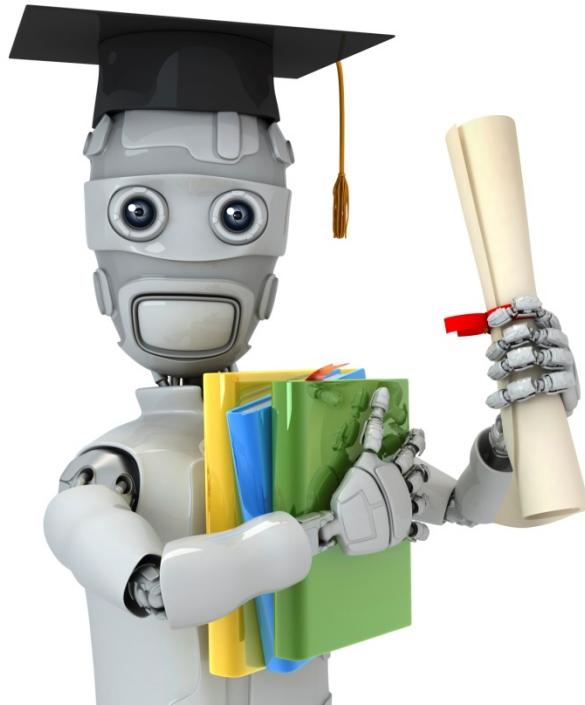
→  $x^{(i)}$  = features of machine  $i$

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

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

...

$$p(x) < \varepsilon$$



Machine Learning

# Anomaly detection

---

## Gaussian distribution

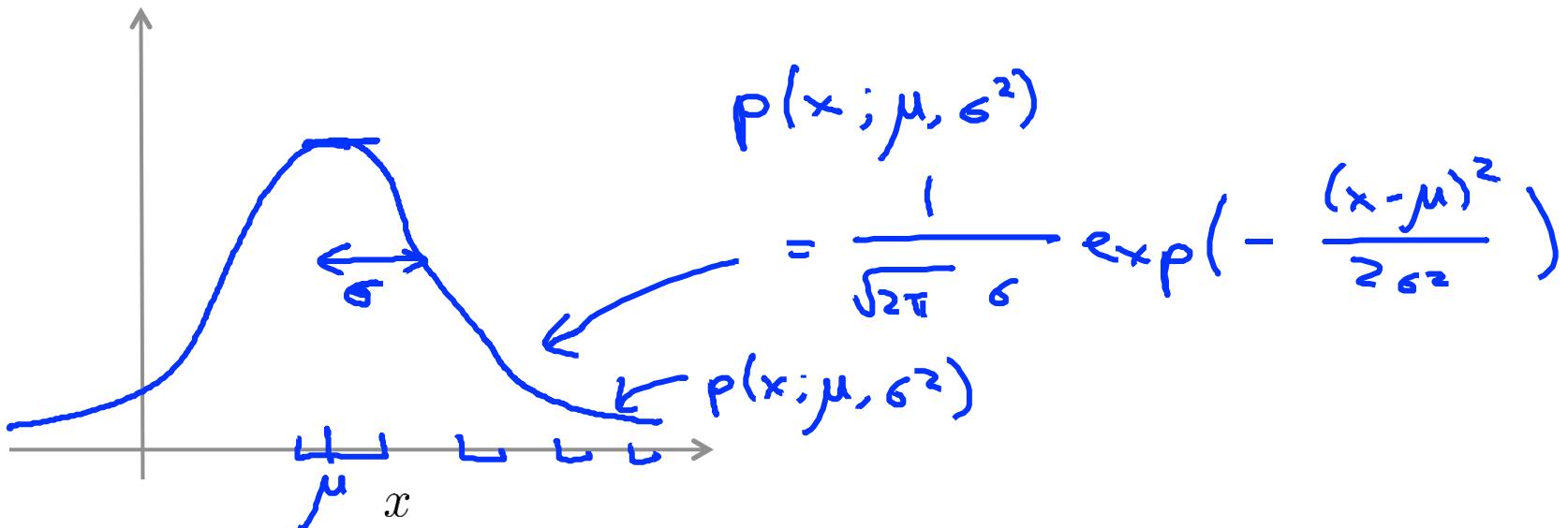
## Gaussian (Normal) distribution

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

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

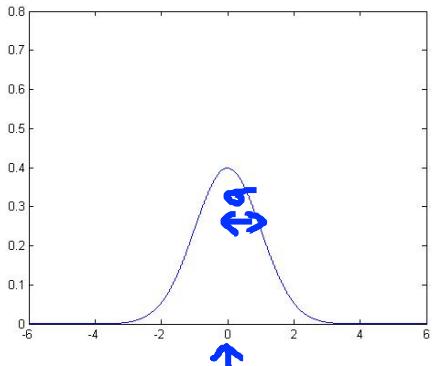
↖ "distributed as"

$\sigma$  standard deviation

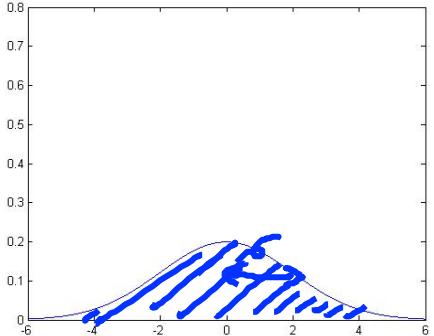


# Gaussian distribution example

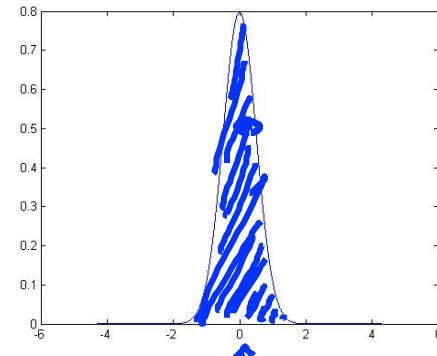
$$\rightarrow \mu = 0, \sigma = 1$$



$$\rightarrow \mu = 0, \sigma = 2$$

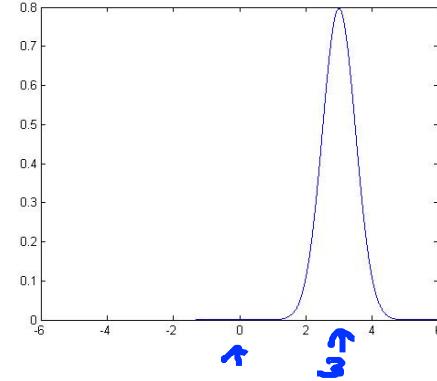


$$\rightarrow \mu = 0, \sigma = 0.5$$



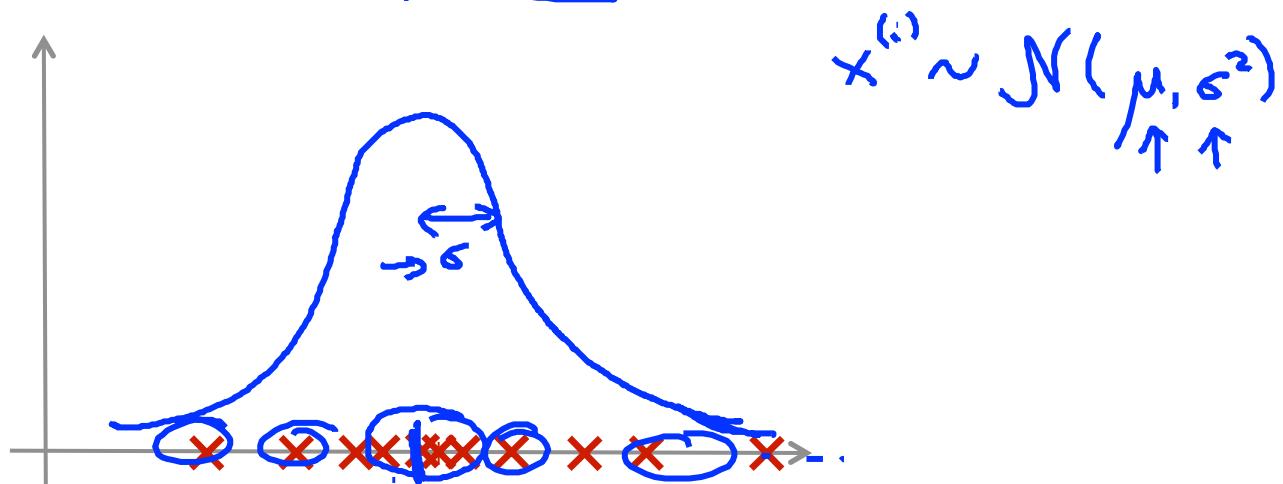
$$\zeta^2 = 0.25$$

$$\rightarrow \mu = 3, \sigma = 0.5$$



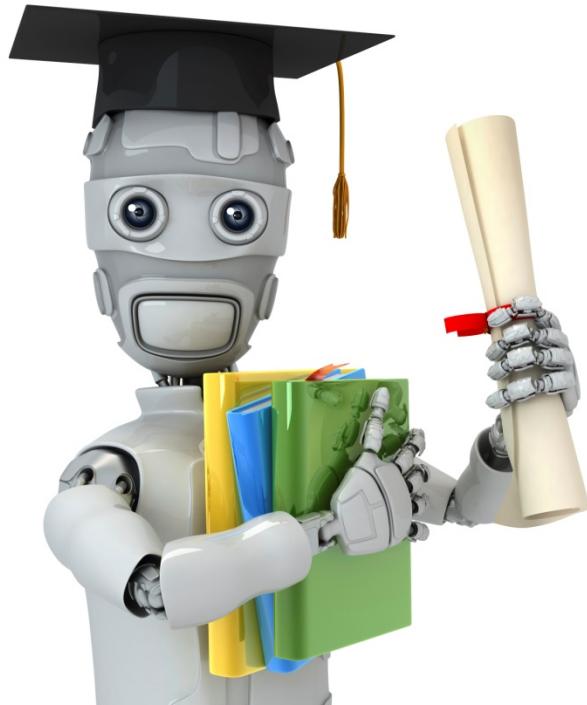
## Parameter estimation

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



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

$$\Rightarrow \hat{\sigma}^2 = \frac{1}{m-1} \sum_{i=1}^m (x^{(i)} - \hat{\mu})^2$$



Machine Learning

# Anomaly detection

---

## Algorithm

## → Density estimation

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

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

→  $p(x)$

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

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

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

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

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

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

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

# 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  $\mu_1, \dots, \mu_n, \sigma_1^2, \dots, \sigma_n^2$

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

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

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

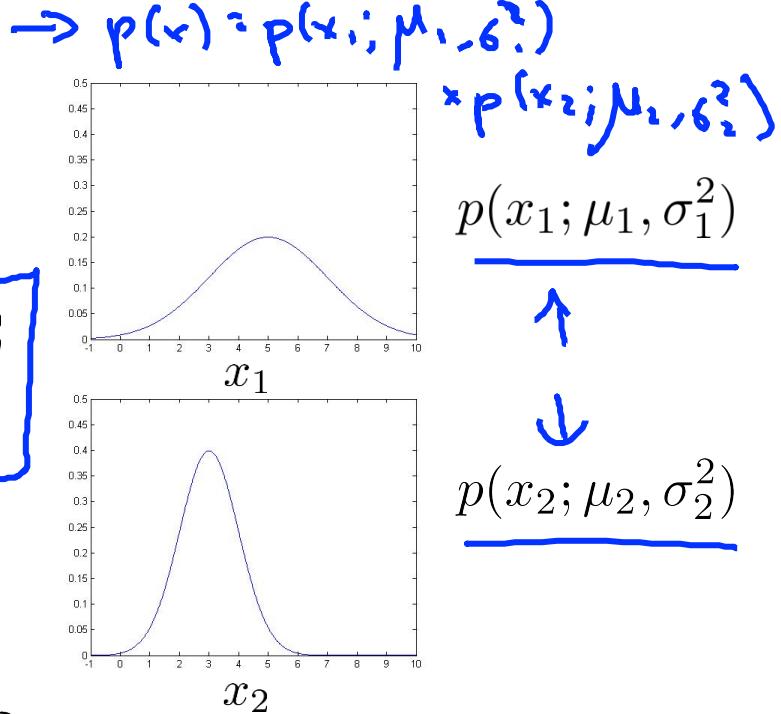
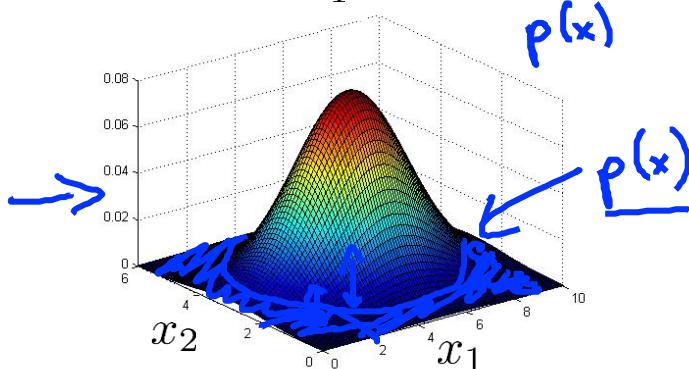
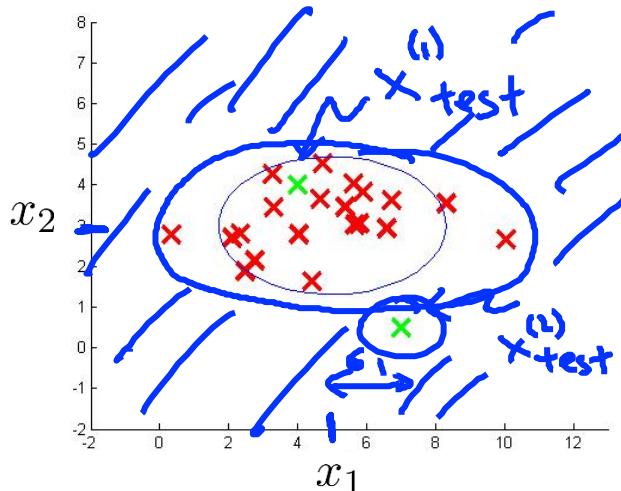
$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix} = \frac{1}{m} \sum_{i=1}^m x^{(i)}$$

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

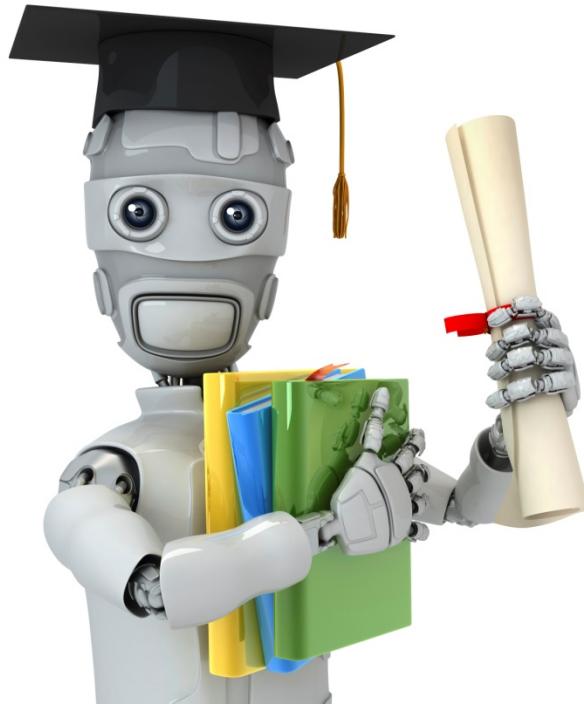
# Anomaly detection example



$$\varepsilon = 0.02$$

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

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



Machine Learning

# Anomaly detection

---

Developing and  
evaluating an anomaly  
detection system

## The importance of real-number evaluation

When developing a learning algorithm (choosing features, etc.), making decisions is much easier if we have a way of evaluating our learning algorithm.

- Assume we have some labeled data, of anomalous and non-anomalous examples. ( $y = 0$  if normal,  $y = 1$  if anomalous).
- Training set:  $x^{(1)}, x^{(2)}, \dots, x^{(m)}$  (assume normal examples/not anomalous)
- Cross validation set:  $(x_{cv}^{(1)}, y_{cv}^{(1)}), \dots, (x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})})$
- Test set:  $(x_{test}^{(1)}, y_{test}^{(1)}), \dots, (x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$

$$y=1$$

## Aircraft engines motivating example

- 10000 good (normal) engines
- 20 flawed engines (anomalous) 2 - 50 y = 1
- 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 ( $y = 1$ )

Alternative:

Training set: 6000 good engines

→ CV: 4000 good engines ( $y = 0$ ), 10 anomalous ( $y = 1$ )

→ Test: 4000 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

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



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

$y=0$

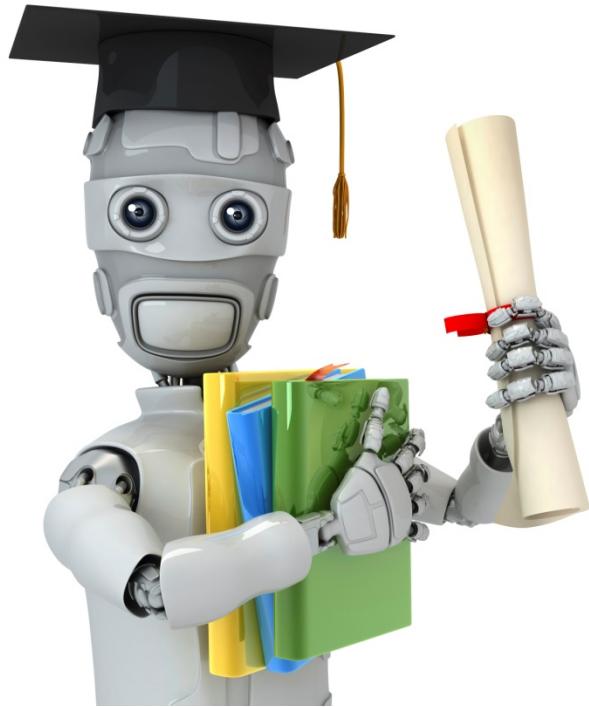
Possible evaluation metrics:

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

CV

Test set

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



Machine Learning

# Anomaly detection

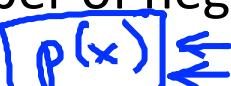
---

Anomaly detection  
vs. supervised  
learning

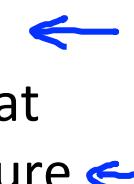
## Anomaly detection

vs.

## Supervised learning

- Very small number of positive examples ( $y = 1$ ). (0-20 is common).
- Large number of negative ( $y = 0$ ) examples. 
- 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.

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. 

Spam 

## Anomaly detection

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

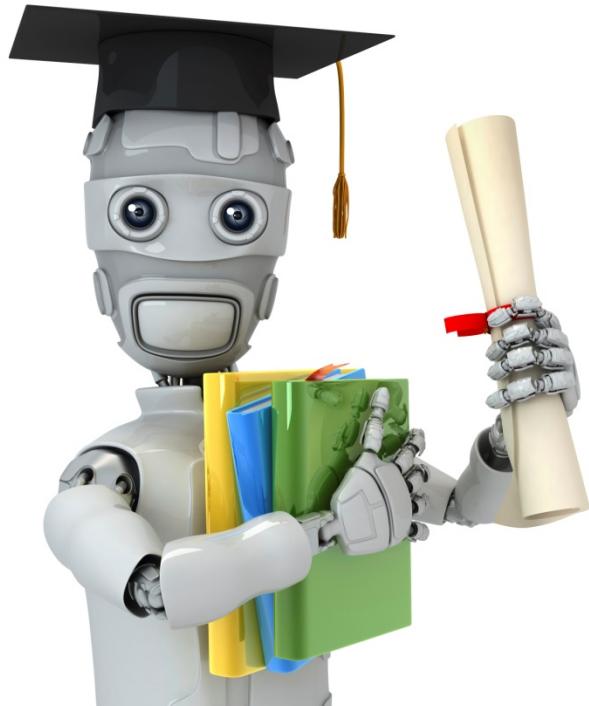
vs.

## Supervised learning

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

:

:



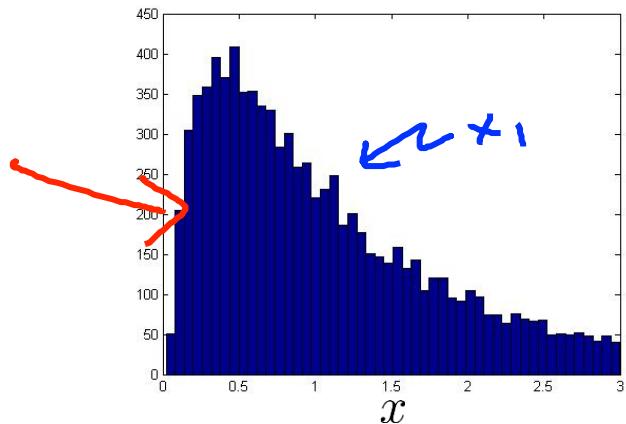
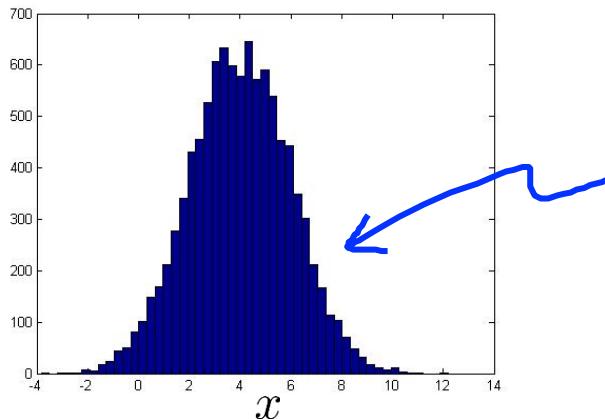
Machine Learning

# Anomaly detection

---

Choosing what features to use

# Non-gaussian features



$p(x_i; \mu_i, \sigma_i^2)$

hist

$x_1 \leftarrow \log(x_1)$

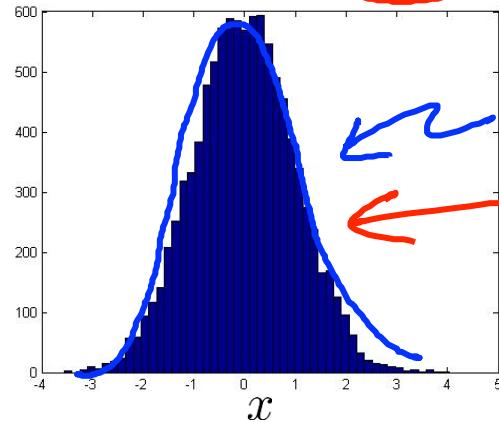
$x_2 \leftarrow \log(x_2 + 1)$

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

$x_4 \leftarrow \frac{x_4}{x_4 + 1} =$

$\log(x_2 + 1)$

Diagram illustrating feature transformation. A red oval encloses four steps: 1.  $x_1 \leftarrow \log(x_1)$ , 2.  $x_2 \leftarrow \log(x_2 + 1)$ , 3.  $x_3 \leftarrow \sqrt{x_3} = x_3^{1/2}$ , and 4.  $x_4 \leftarrow \frac{x_4}{x_4 + 1} =$ . Blue arrows point from the original histograms to these transformed features.

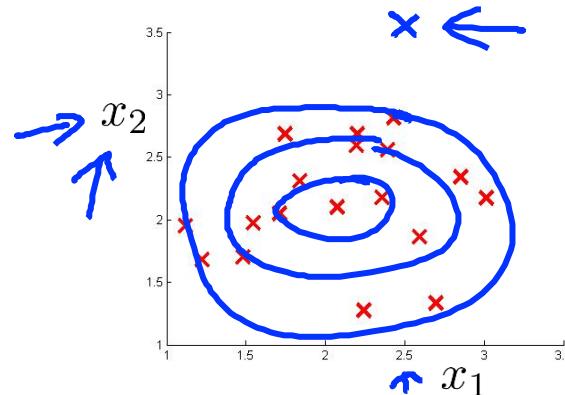
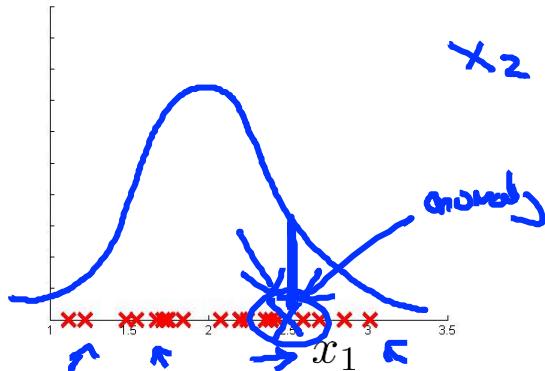


## → Error analysis for anomaly detection

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

Most common problem:

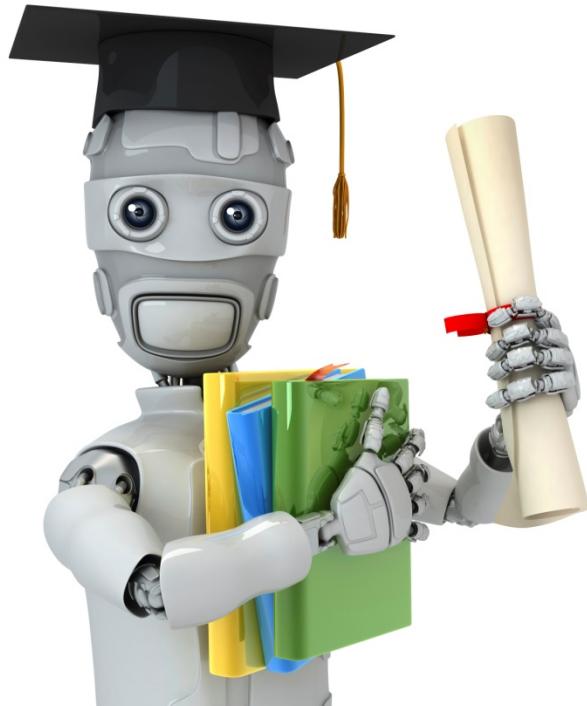
$p(x)$  is comparable (say, both large) for normal and anomalous examples



- 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 ←

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

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



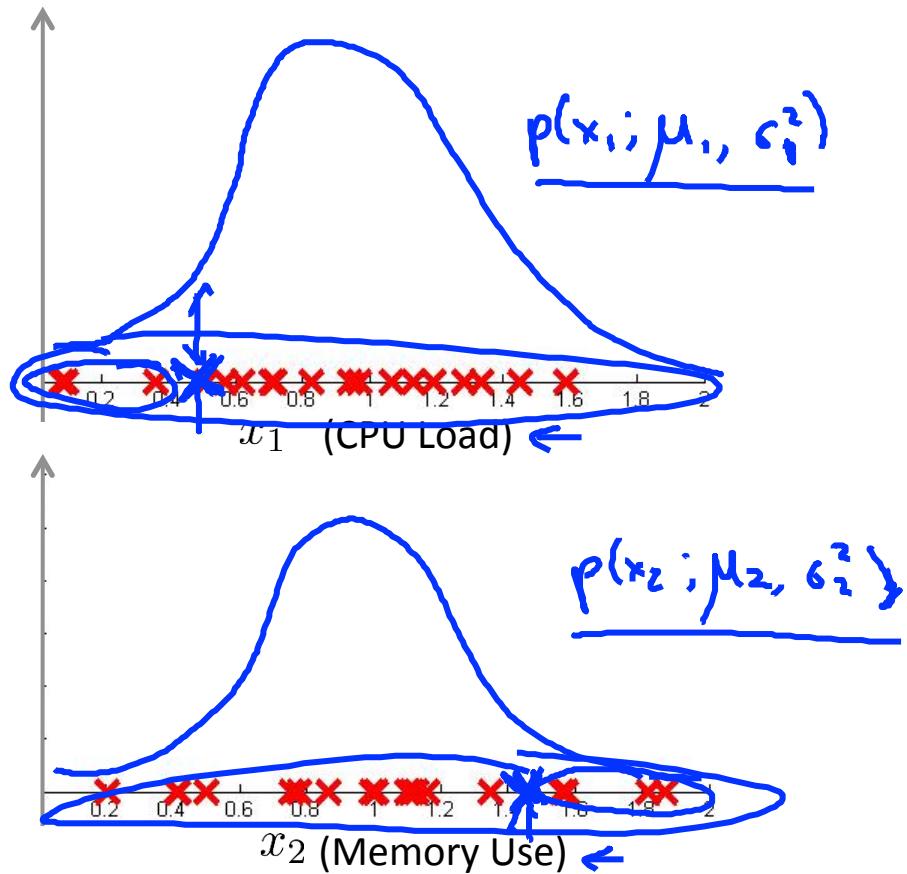
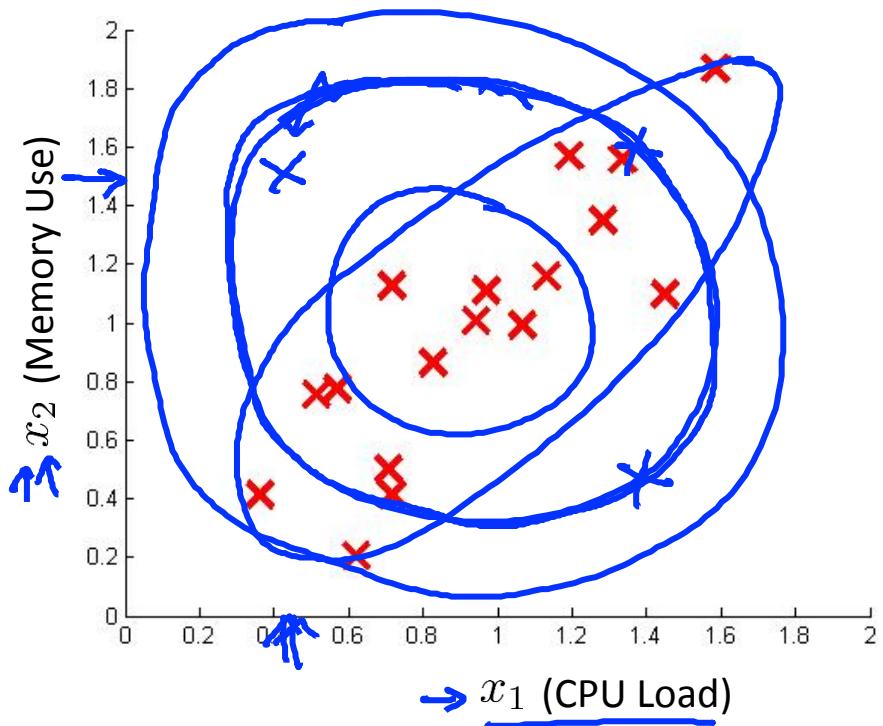
Machine Learning

# Anomaly detection

---

Multivariate  
Gaussian distribution

# 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)

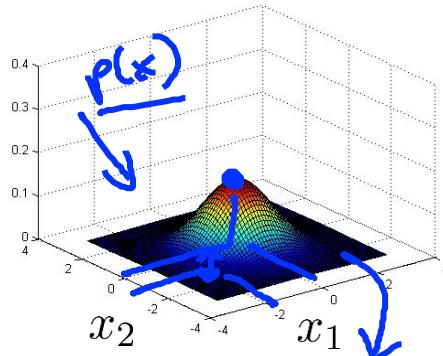
$$p(x; \mu, \Sigma) =$$

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

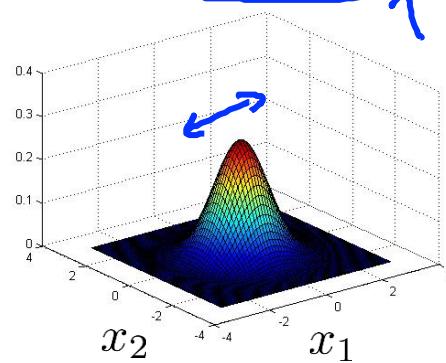
$|\Sigma| = \text{determinant of } \Sigma \quad | \det(\text{Sigma})$

# Multivariate Gaussian (Normal) examples

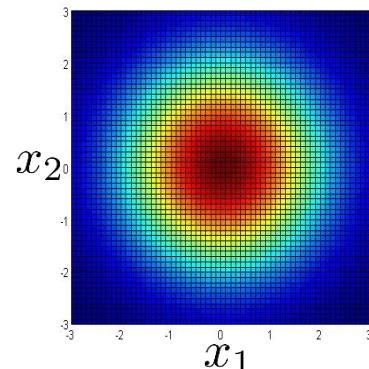
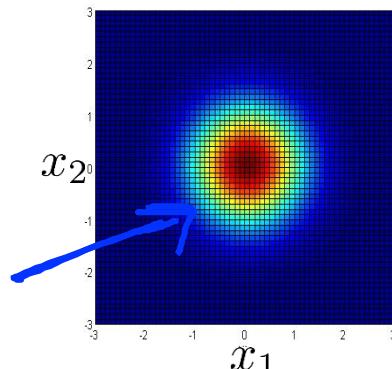
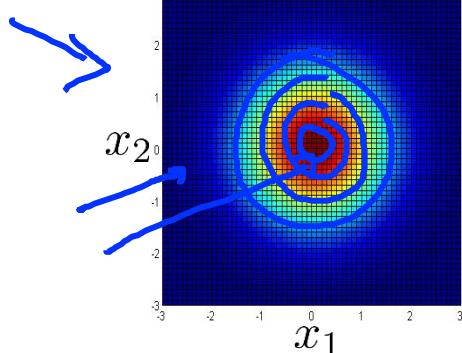
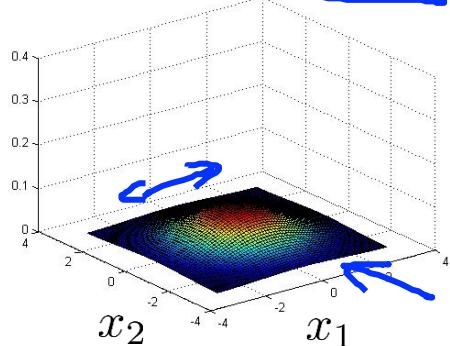
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix}$$

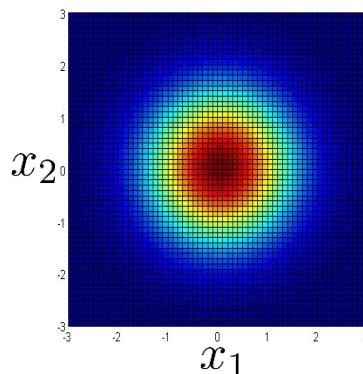
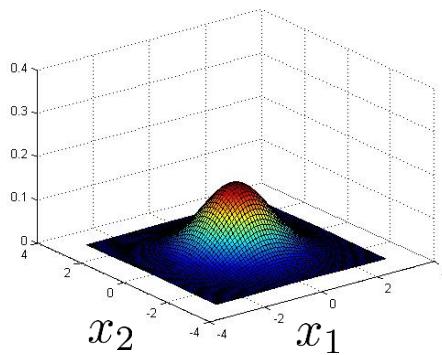


$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

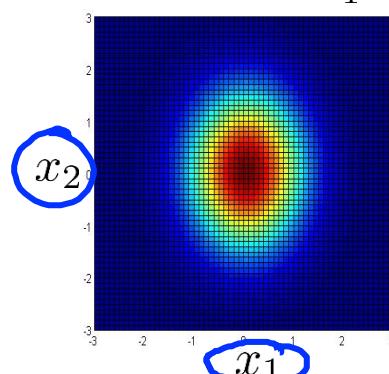
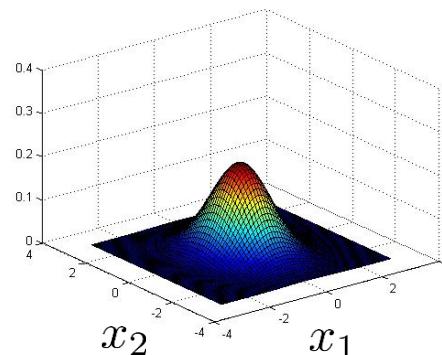


# Multivariate Gaussian (Normal) examples

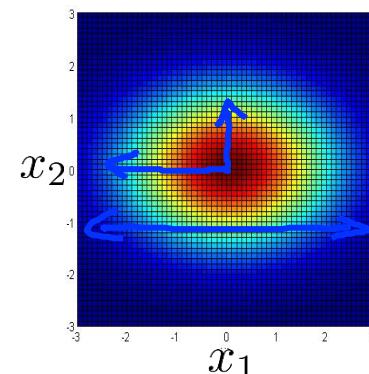
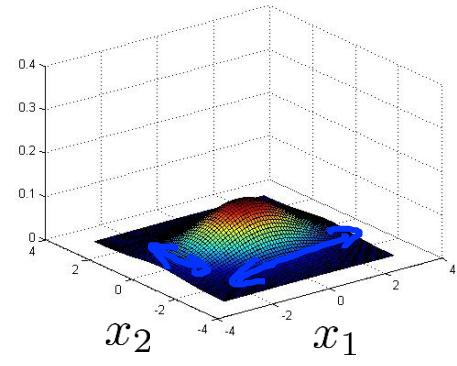
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 1 \end{bmatrix}$$

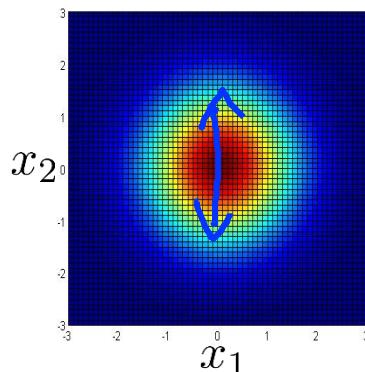
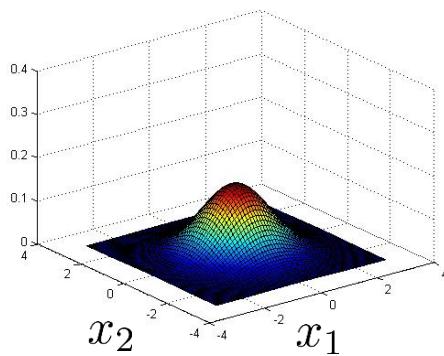


$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

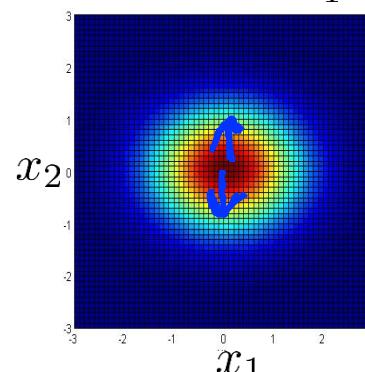
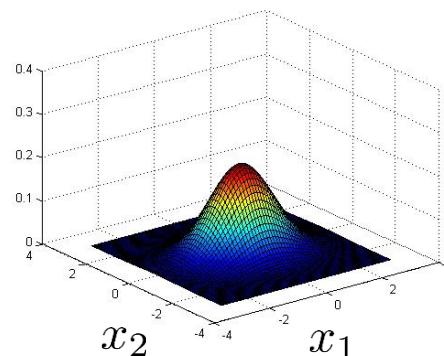


# Multivariate Gaussian (Normal) examples

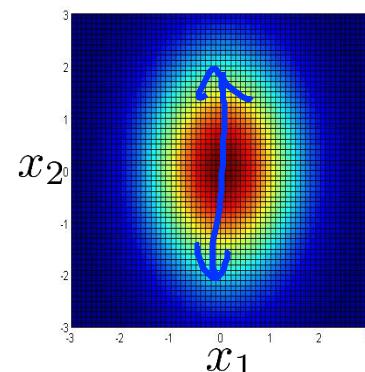
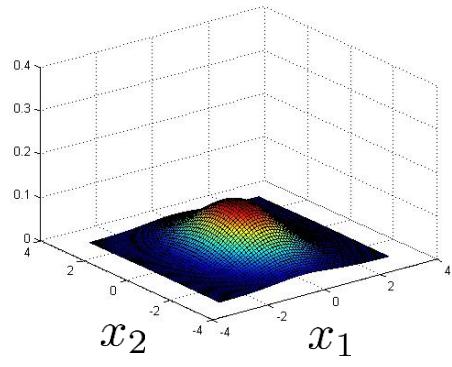
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 0.6 \end{bmatrix}$$

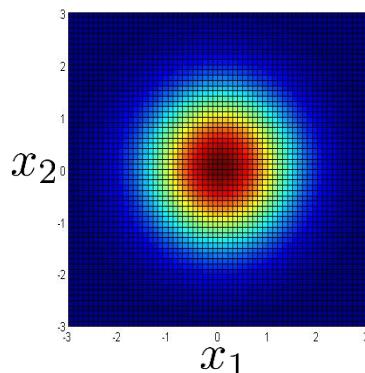
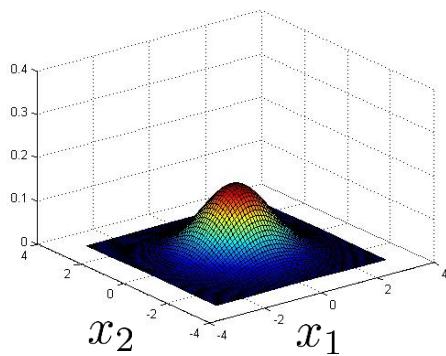


$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

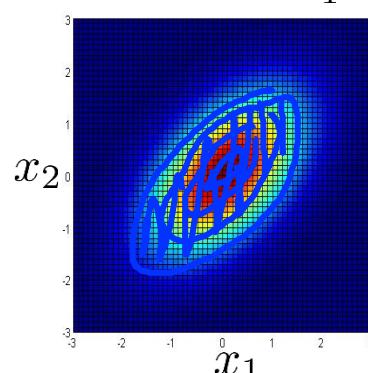
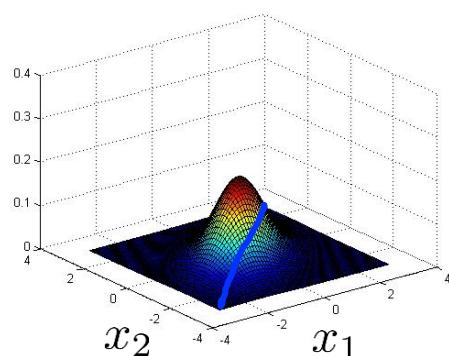


# Multivariate Gaussian (Normal) examples

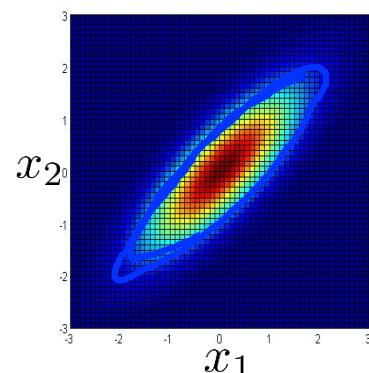
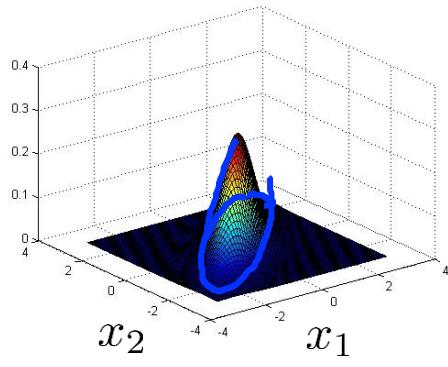
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

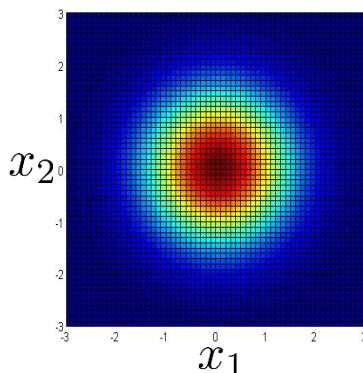
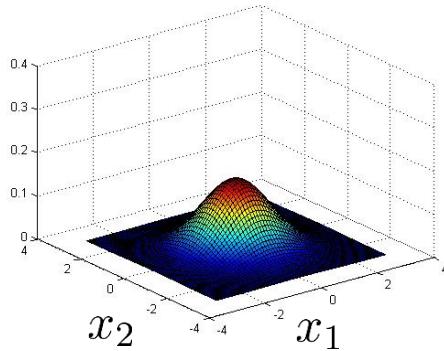


$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

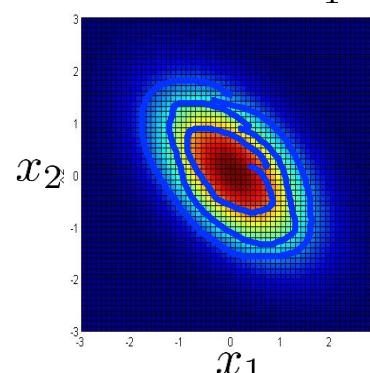
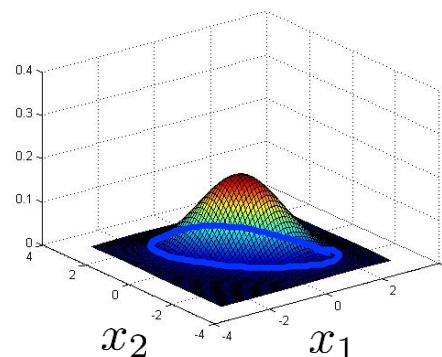


# Multivariate Gaussian (Normal) examples

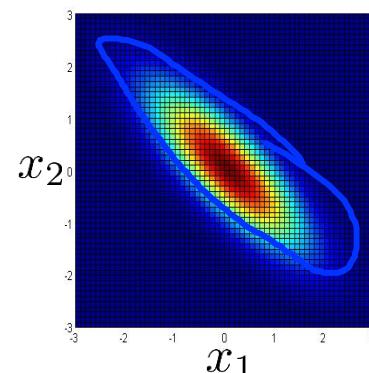
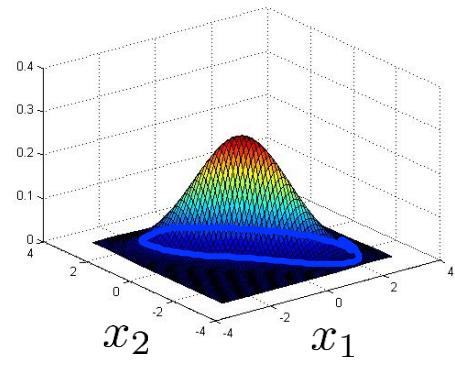
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$$

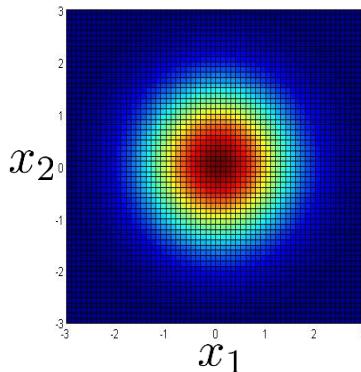
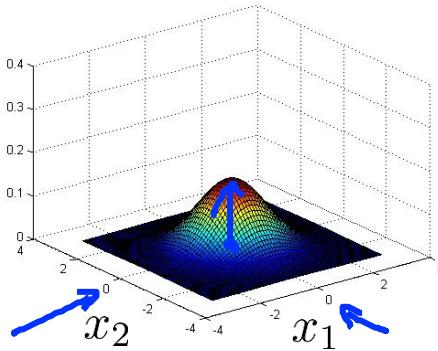


$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}$$

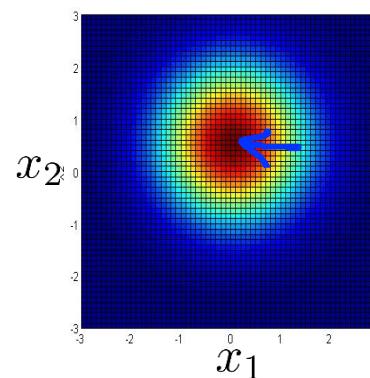
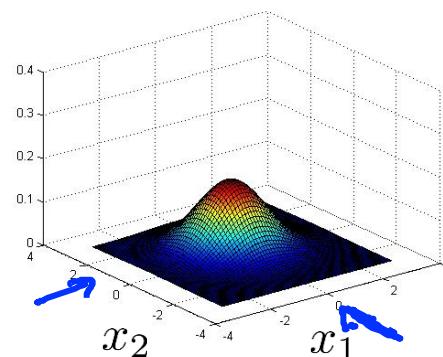


# Multivariate Gaussian (Normal) examples

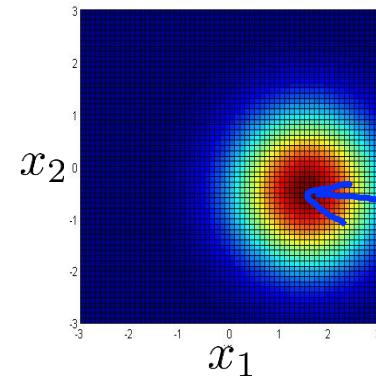
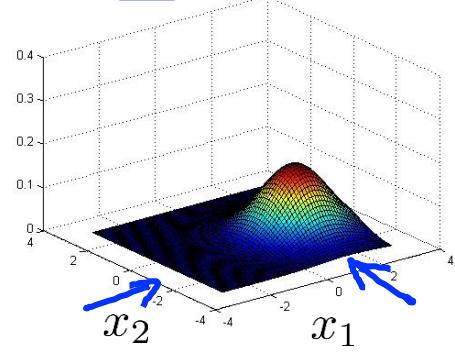
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

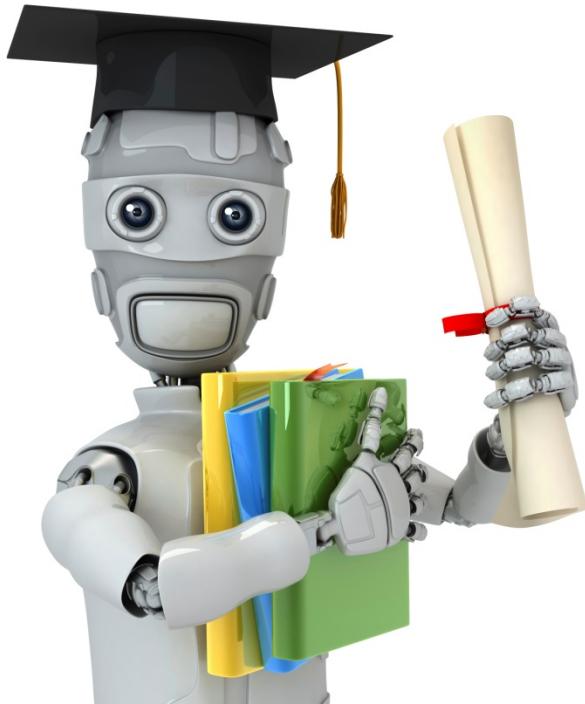


$$\mu = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$\mu = \begin{bmatrix} 1.5 \\ -0.5 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$





Machine Learning

# Anomaly detection

---

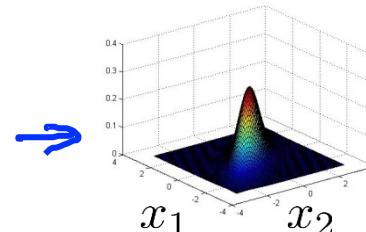
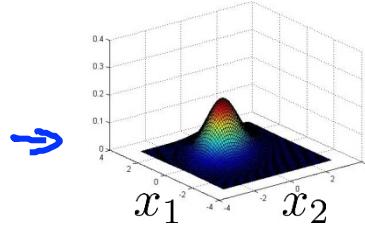
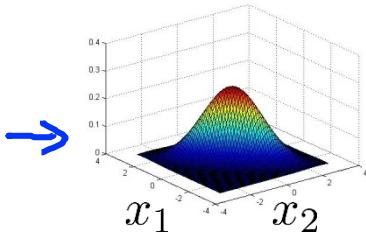
Anomaly detection using  
the multivariate  
Gaussian distribution

# Multivariate Gaussian (Normal) distribution

Parameters  $\mu, \Sigma$

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

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



Parameter fitting:

Given training set  $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$

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

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

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

# Anomaly detection with the multivariate Gaussian

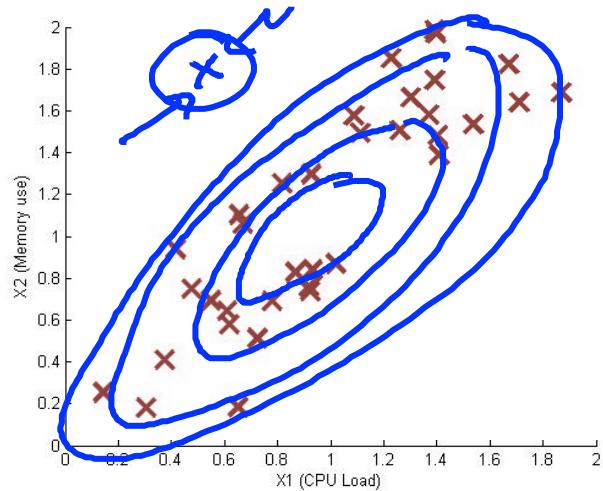
1. Fit model  $p(x)$  by setting

$$\left[ \begin{array}{l} \mu = \frac{1}{m} \sum_{i=1}^m x^{(i)} \\ \Sigma = \frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu)(x^{(i)} - \mu)^T \end{array} \right]$$

2. Given a new example  $x$ , compute

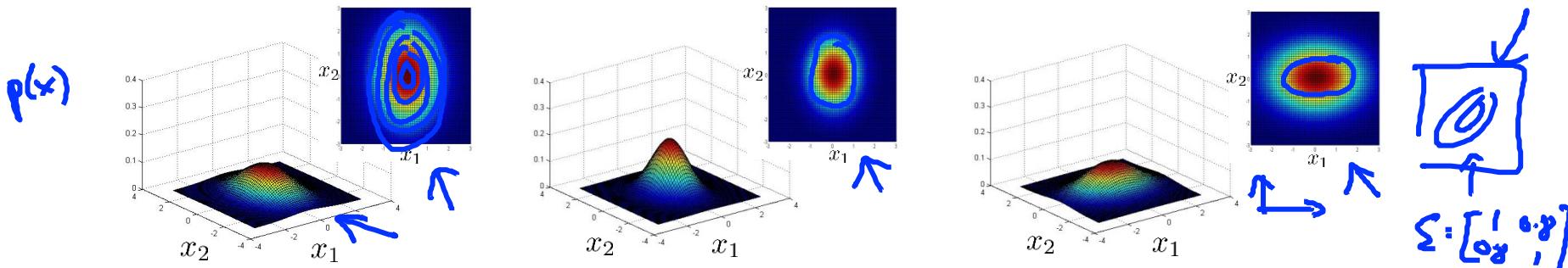
$$p(x) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right)$$

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



## 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

$$\Sigma = \begin{bmatrix} \dots & & \\ & \ddots & \\ & & \dots \end{bmatrix}$$

## → 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  $n=10,000, n=100,000$ )
  - 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

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

$$\underline{\Sigma^{-1}}$$

Computationally more expensive

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

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 + x_5 \\ x_5 \end{bmatrix}$$

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

$$\underline{m \geq n}$$