



Machine Learning

# Anomaly detection

---

Problem  
motivation

# Anomaly detection example

Aircraft engine features:

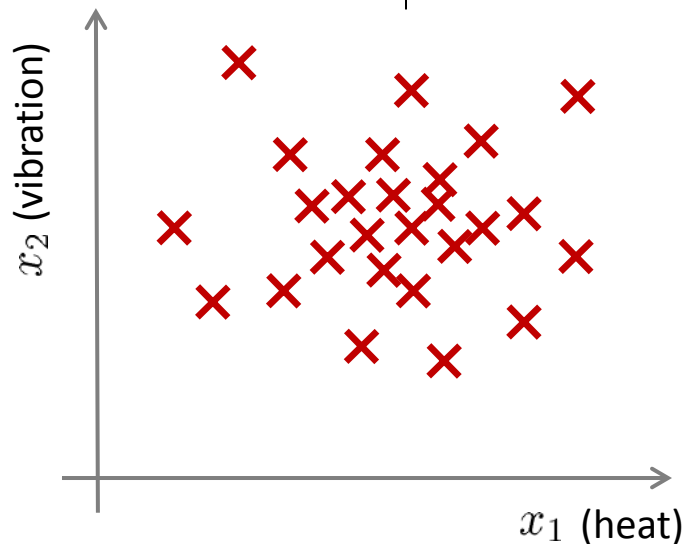
$x_1$  = heat generated

$x_2$  = vibration intensity

...

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

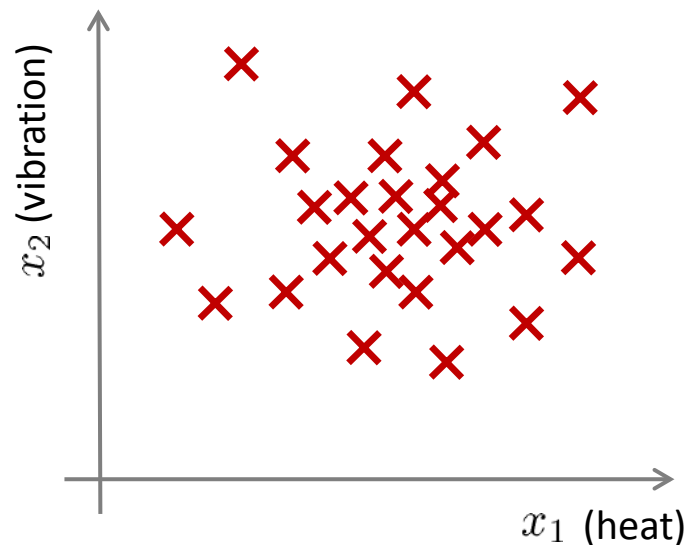
New engine:  $x_{test}$



# Density estimation

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

Is  $x_{test}$  anomalous?



# 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  $p(x) < \varepsilon$

Manufacturing

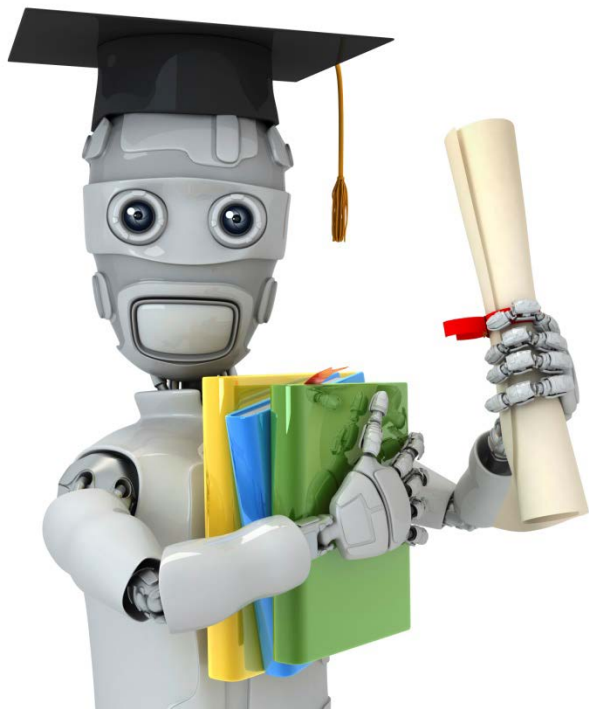
Monitoring computers in a data center.

$x^{(i)}$  = 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.

...



Machine Learning

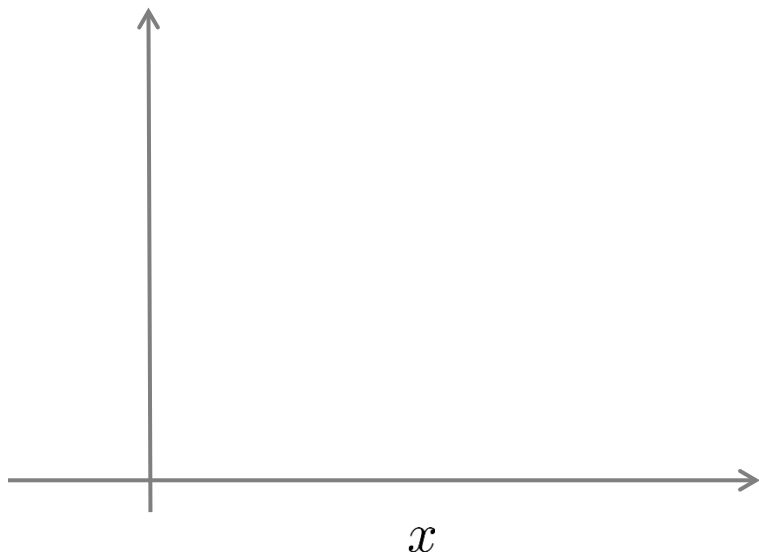
# Anomaly detection

---

## Gaussian distribution

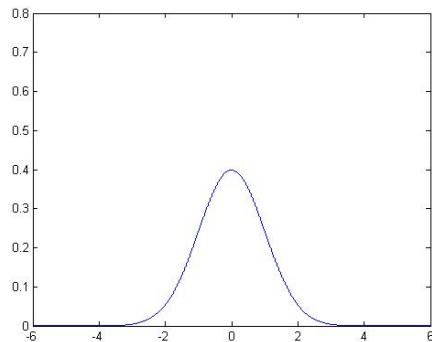
## Gaussian (Normal) distribution

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

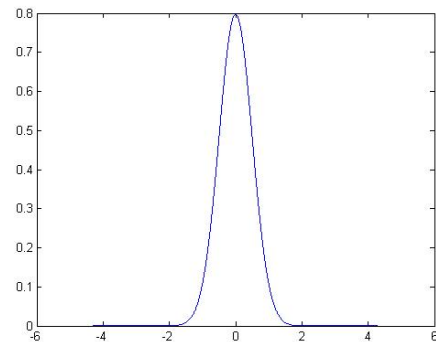


# Gaussian distribution example

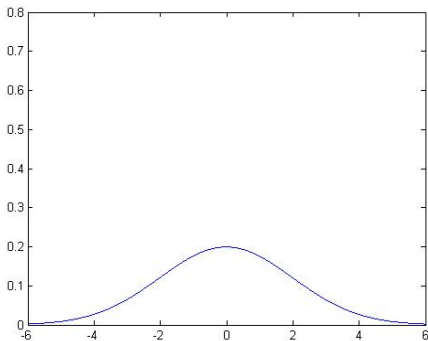
$$\mu = 0, \sigma = 1$$



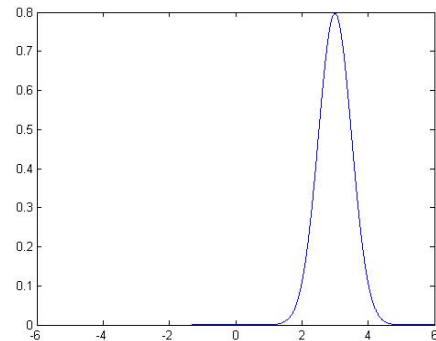
$$\mu = 0, \sigma = 0.5$$



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

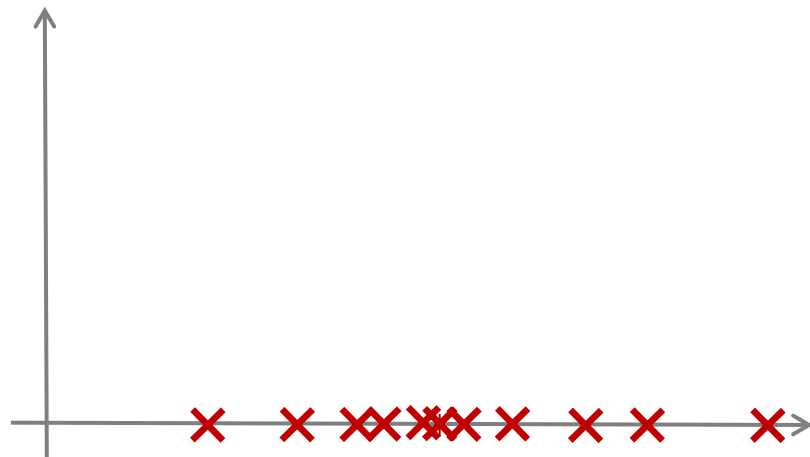


$$\mu = 3, \sigma = 0.5$$



# Parameter estimation

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







Machine Learning

# Anomaly detection

---

# Algorithm

# Density estimation

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

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

# Anomaly detection algorithm

1. Choose features  $x_i$  that you think might be indicative of anomalous examples.
2. Fit parameters  $\mu_1, \dots, \mu_n, \sigma_1^2, \dots, \sigma_n^2$

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

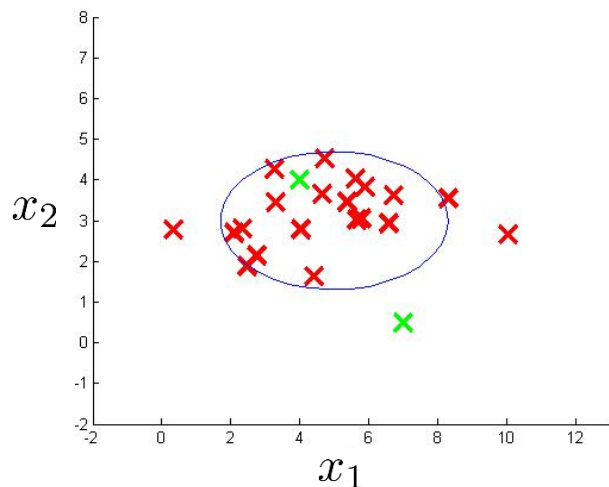
$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)} - \mu_j)^2$$

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

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

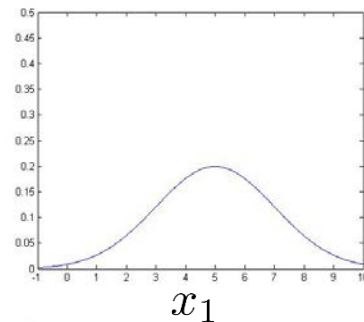
Anomaly if  $p(x) < \varepsilon$

# Anomaly detection example

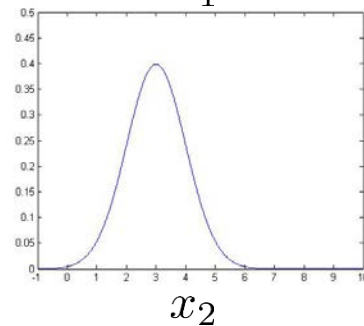


$$\mu_1 = 5, \sigma_1 = 2$$

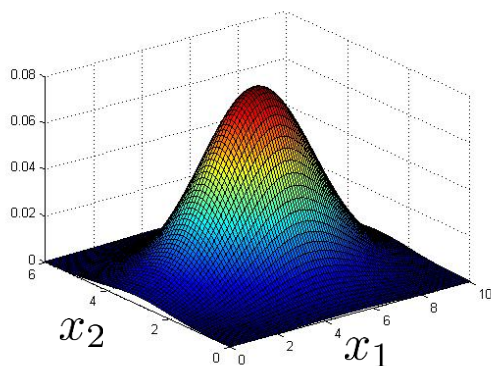
$$\mu_2 = 3, \sigma_2 = 1$$



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



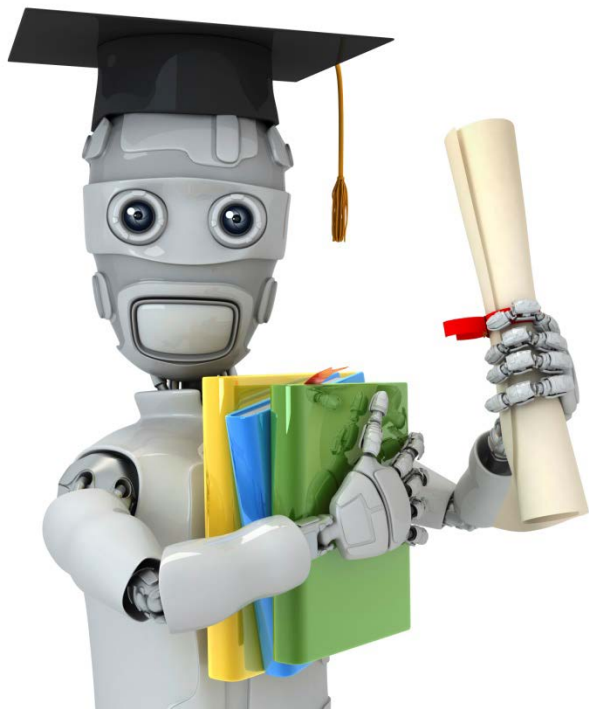
$$p(x_2; \mu_2, \sigma_2^2)$$



$$\varepsilon = 0.02$$

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

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



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:  $(x_{test}^{(1)}, y_{test}^{(1)}), \dots, (x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$

## Aircraft engines motivating example

10000 good (normal) engines

20 flawed engines (anomalous)

Training set: 6000 good engines

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

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

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  $\varepsilon$





Machine Learning

# Anomaly detection

---

Anomaly detection  
vs. supervised  
learning

## Anomaly detection

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.

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
- Manufacturing (e.g. aircraft engines)
- Monitoring machines in a data center

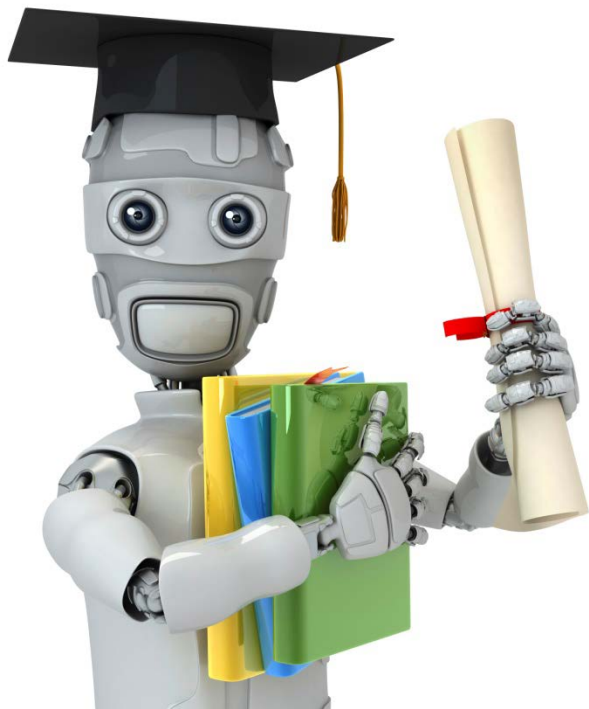
⋮

vs.

## **Supervised learning**

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

⋮



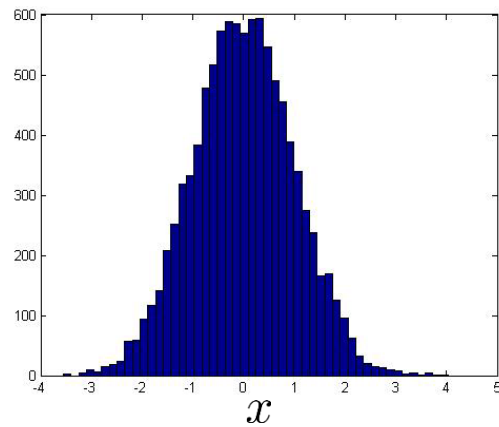
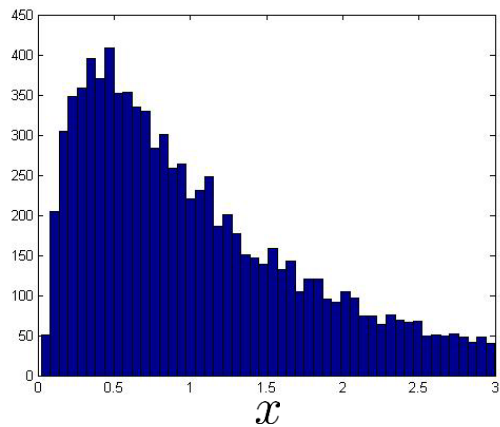
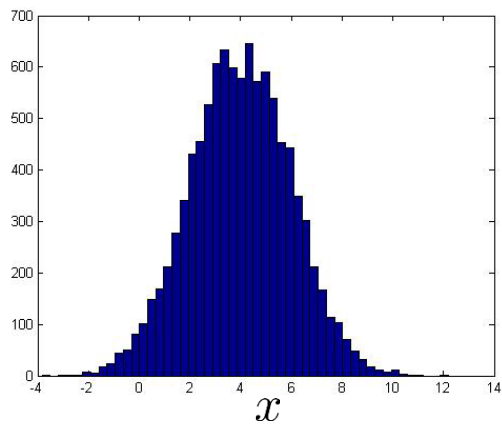
Machine Learning

# Anomaly detection

---

Choosing what  
features to use

# Non-gaussian features



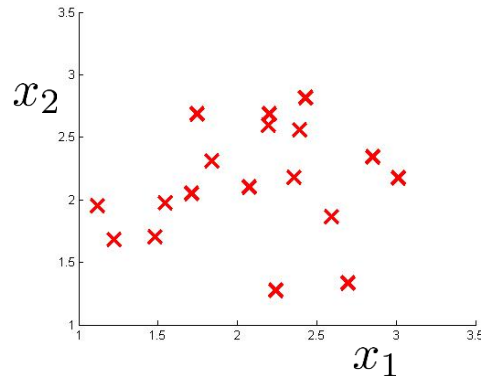
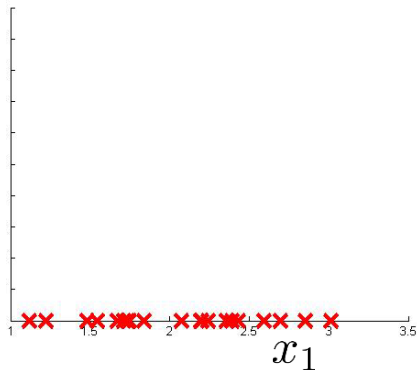
# Error analysis for anomaly detection

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

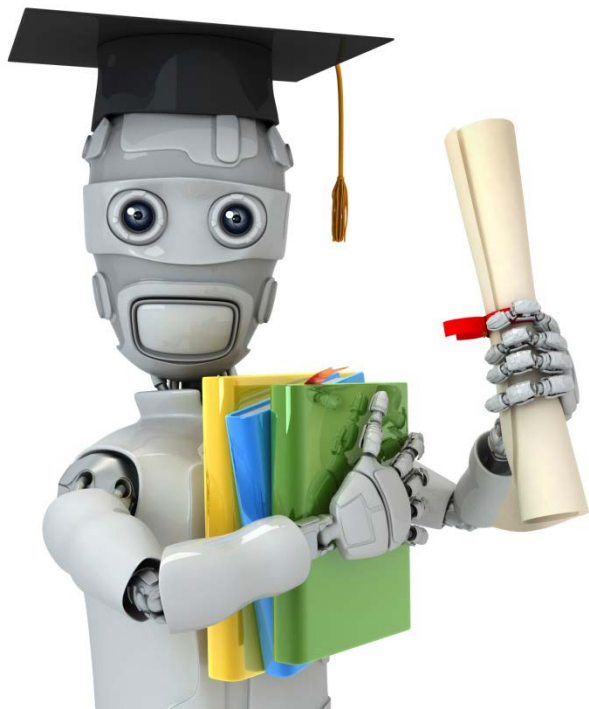
Choose features that might take on unusually large or small values in the event of an anomaly.

$x_1$  = memory use of computer

$x_2$  = number of disk accesses/sec

$x_3$  = CPU load

$x_4$  = network traffic



Machine Learning

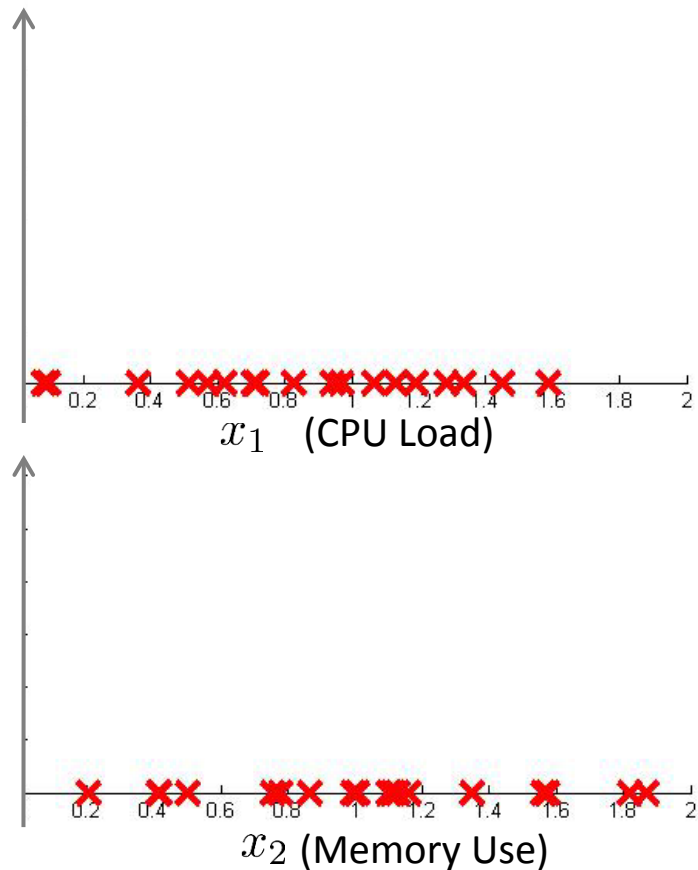
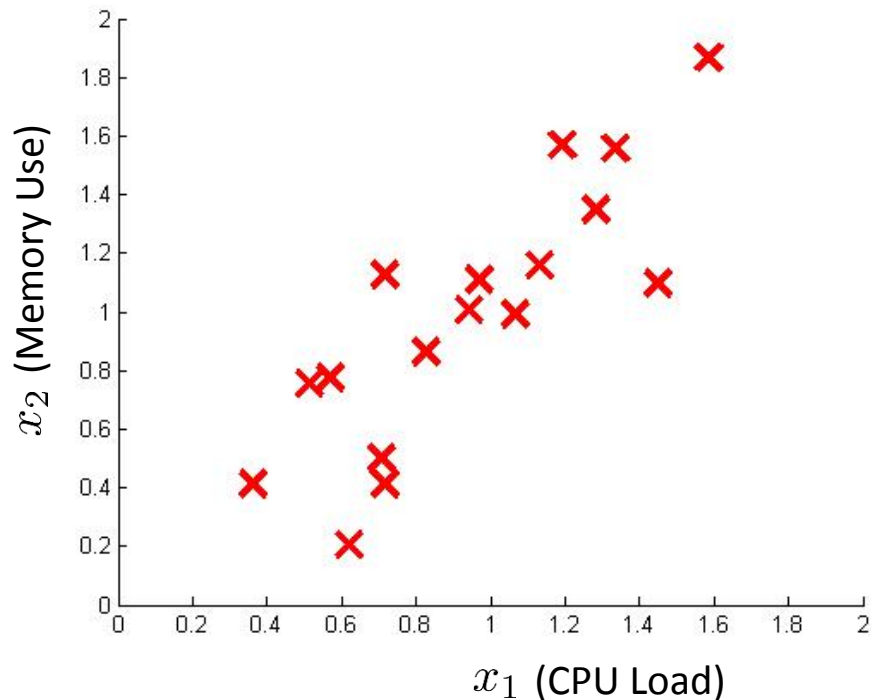
# Anomaly detection

---

Multivariate  
Gaussian distribution



# Motivating example: Monitoring machines in a data center



# Multivariate Gaussian (Normal) distribution

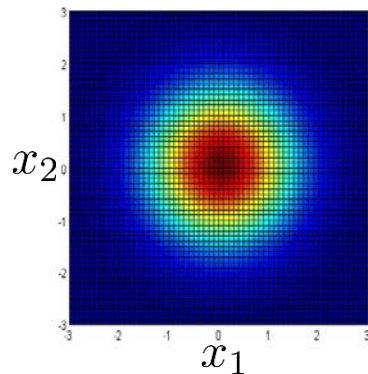
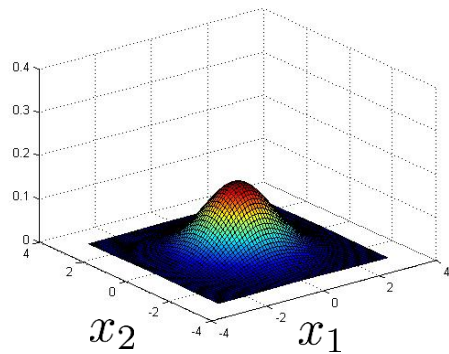
$x \in \mathbb{R}^n$ . Don't model  $p(x_1), p(x_2), \dots$ , etc. separately.

Model  $p(x)$  all in one go.

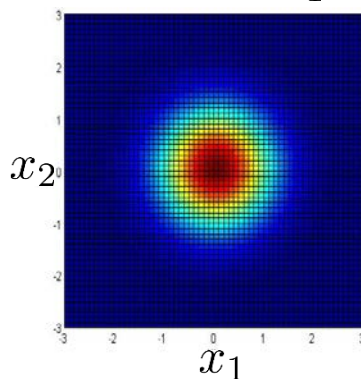
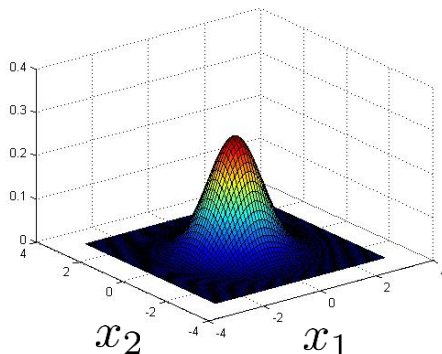
Parameters:  $\mu \in \mathbb{R}^n, \Sigma \in \mathbb{R}^{n \times n}$  (covariance matrix)

# Multivariate Gaussian (Normal) examples

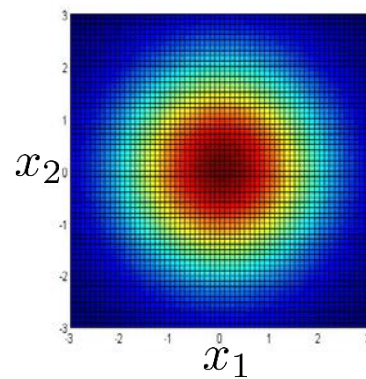
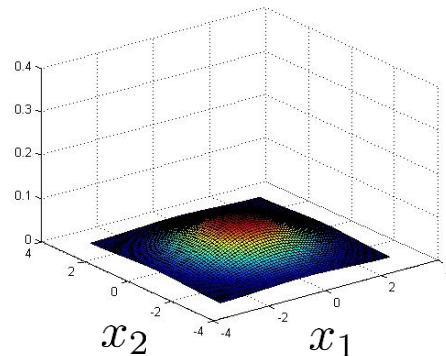
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



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

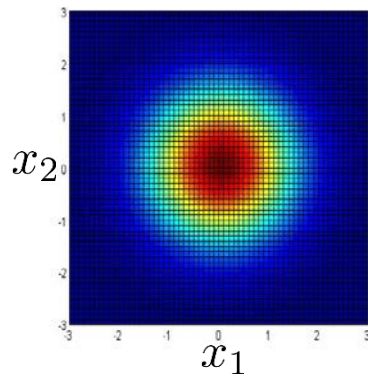
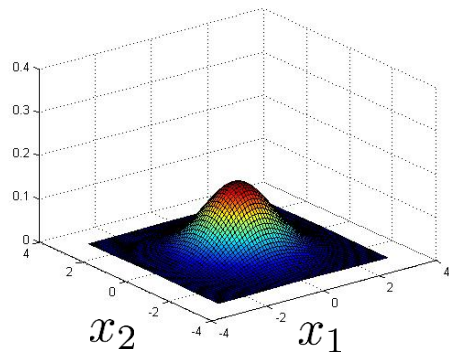


$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

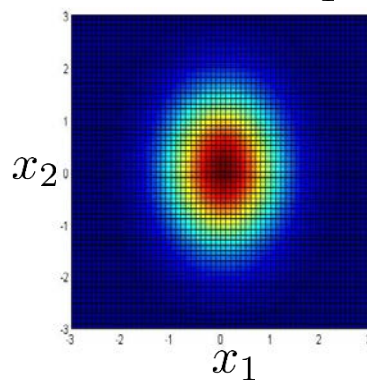


# Multivariate Gaussian (Normal) examples

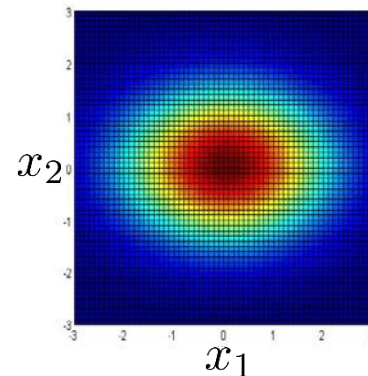
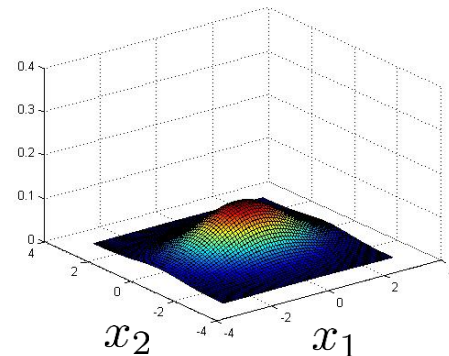
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



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

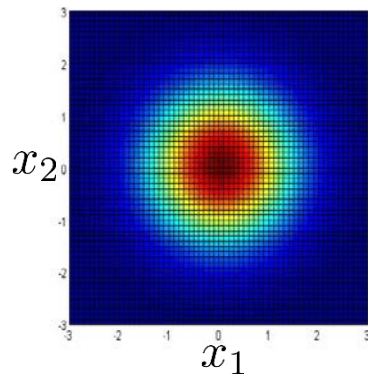
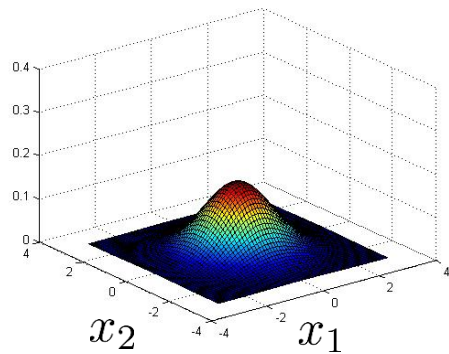


$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

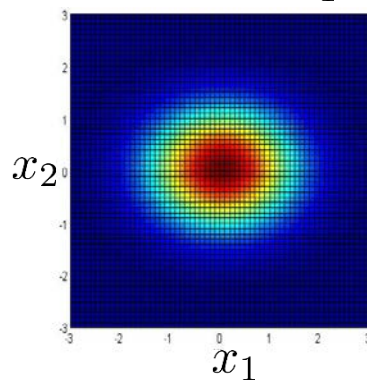


# Multivariate Gaussian (Normal) examples

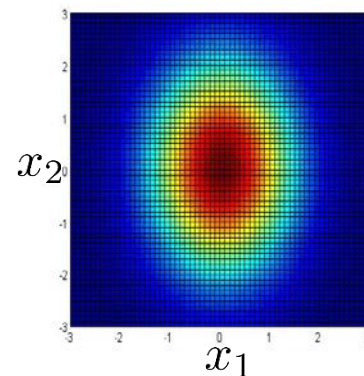
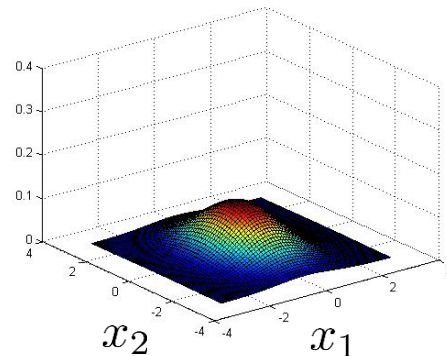
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



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

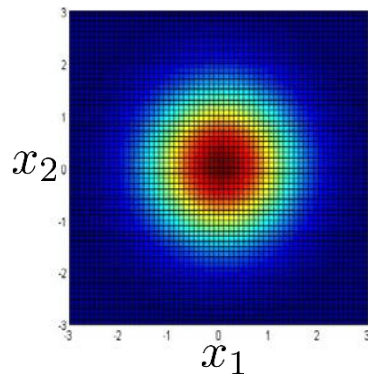
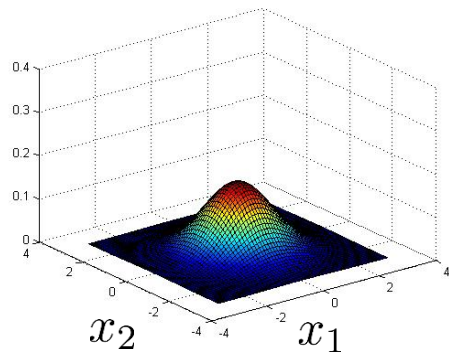


$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

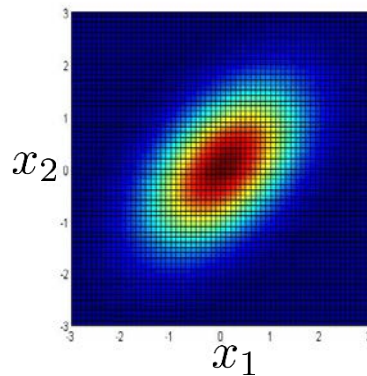
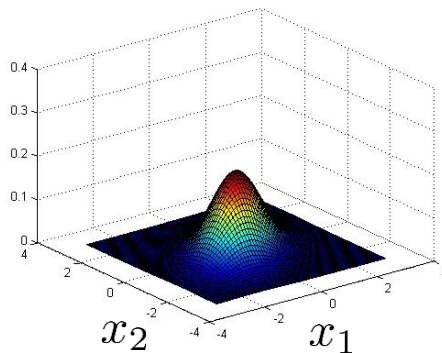


# Multivariate Gaussian (Normal) examples

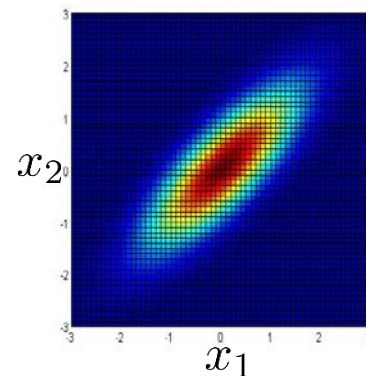
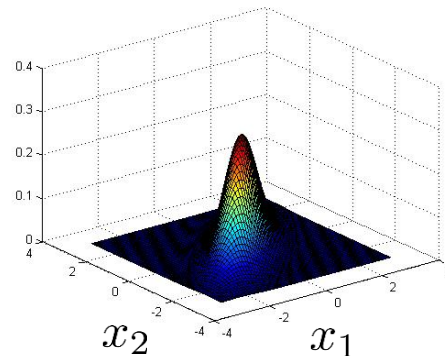
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



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



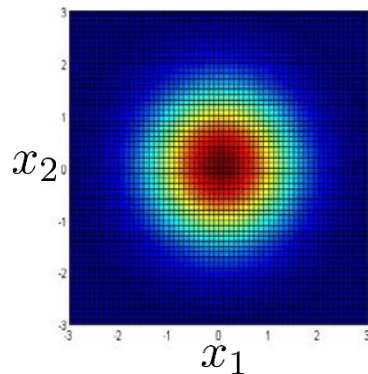
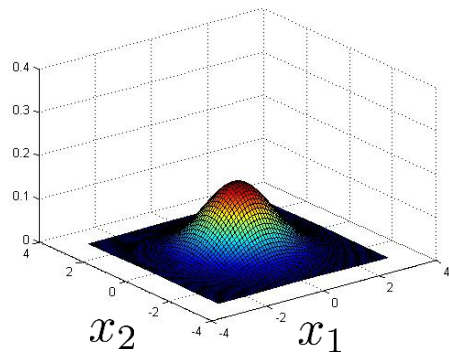
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$



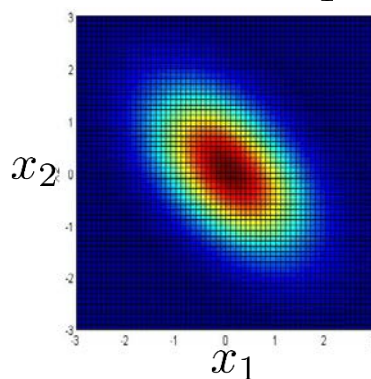
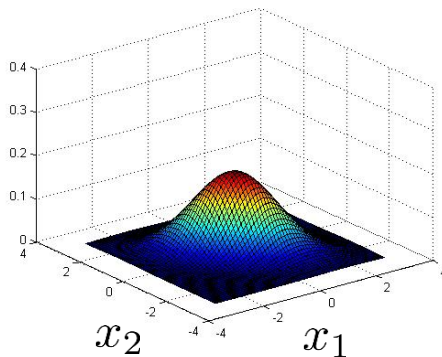


# Multivariate Gaussian (Normal) examples

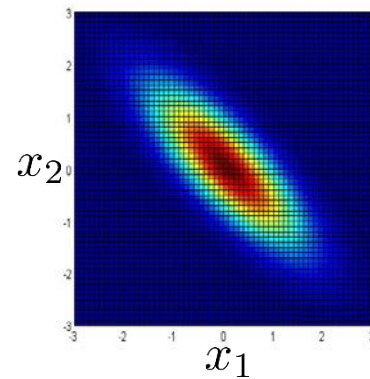
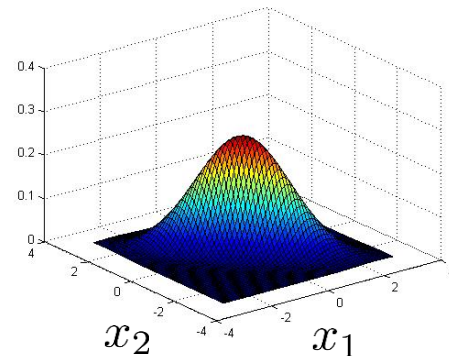
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



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

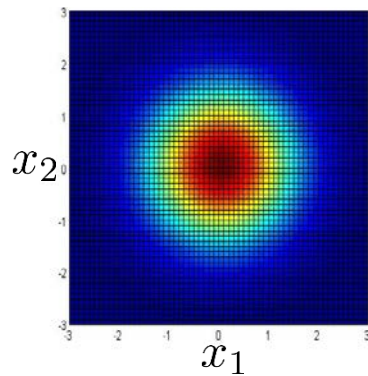
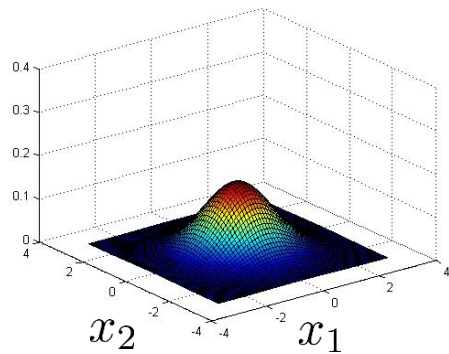


$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}$$

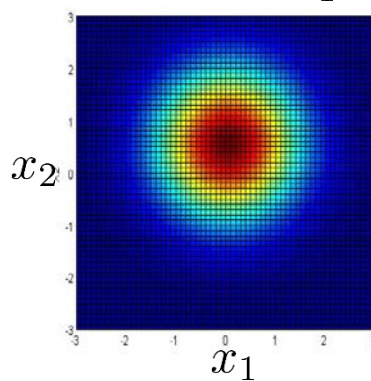
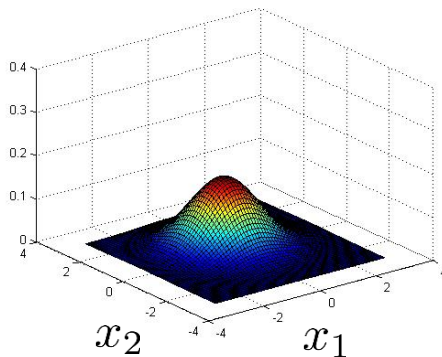


# Multivariate Gaussian (Normal) examples

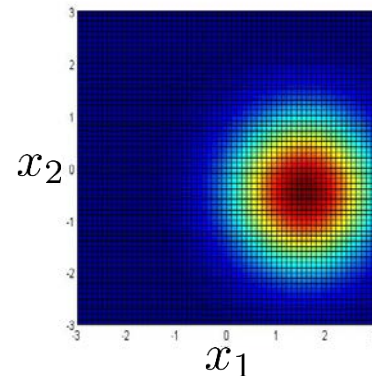
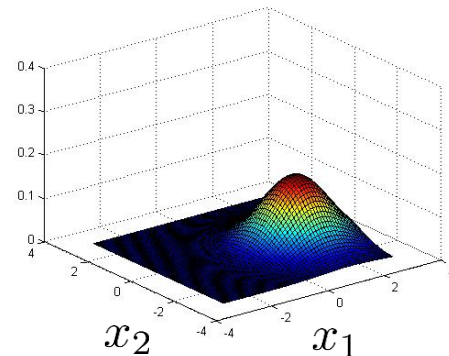
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



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



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







Machine Learning

# Anomaly detection

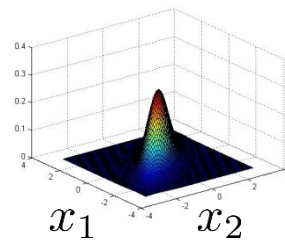
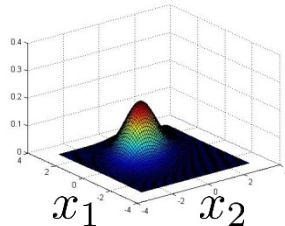
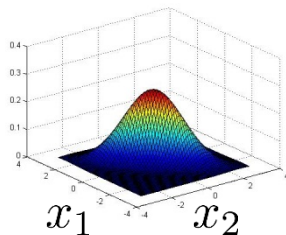
---

Anomaly detection using  
the multivariate  
Gaussian distribution

# Multivariate Gaussian (Normal) distribution

Parameters  $\mu, \Sigma$

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



Parameter fitting:

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

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

# Anomaly detection with the multivariate Gaussian

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

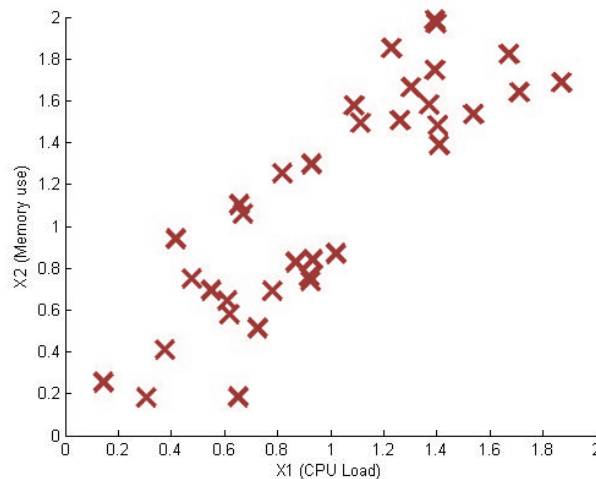
$$\mu = \frac{1}{m} \sum_{i=1}^m x^{(i)}$$

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

2. Given a new example  $x$ , compute

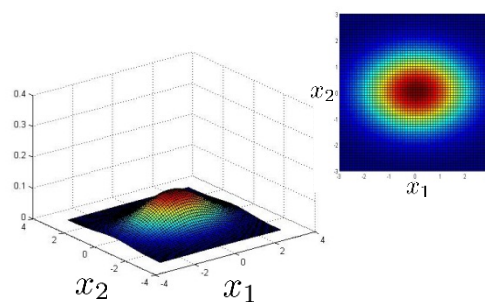
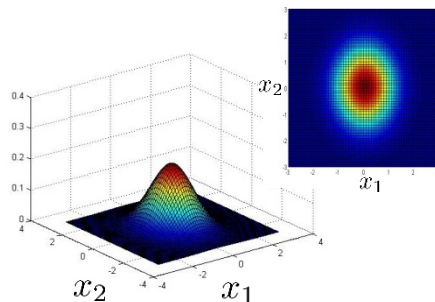
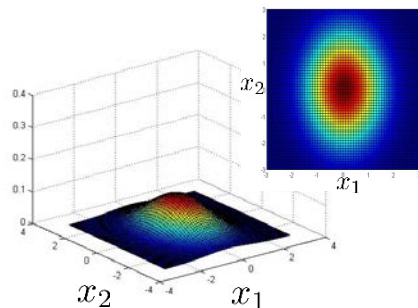
$$p(x) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right)$$

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

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

where

## Original model

vs.

## Multivariate Gaussian

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

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

Computationally cheaper (alternatively, scales better to large  $n$ )

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

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

Automatically captures correlations between features

Computationally more expensive

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