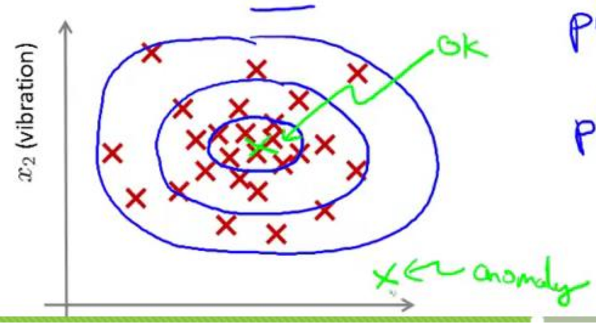
异常检测问题 Anomaly detection

Density estimation

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

Model $p(x)$.



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

$p(x_{test}) \geq \varepsilon \rightarrow$ 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 $p(x) < \varepsilon$

x_1
x_2
x_3
x_4

$p(x)$

- 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.

    ...

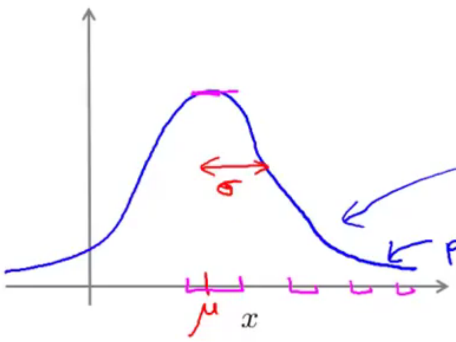
$p(x) < \varepsilon$

高斯分布　正态分布

Gaussian (Normal) distribution

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

$x \sim \mathcal{N}(\mu, \sigma^2)$ "distributed as" σ standard deviation

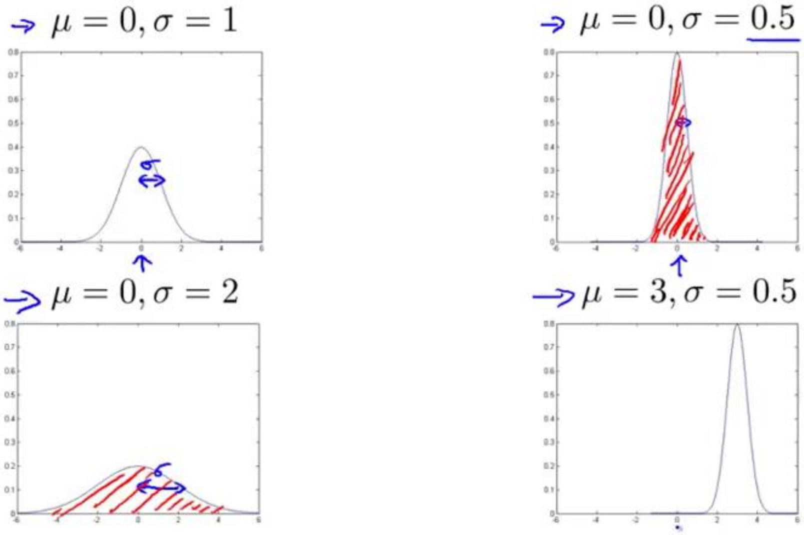


$p(x; \mu, \sigma^2)$

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

$p(x; \mu, \sigma^2)$

Gaussian distribution example

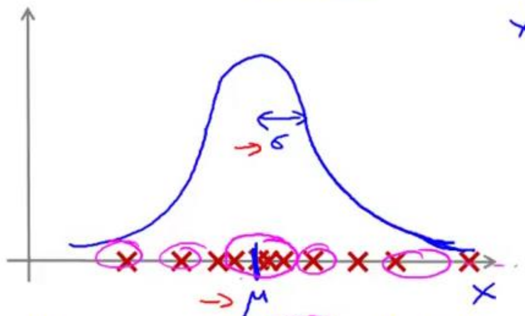


$\sigma^2 = 0.25$

Parameter estimation

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

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



$$\mu = \frac{1}{m} \sum_{i=1}^{m} x^{(i)} \qquad \sigma^2 = \boxed{\frac{1}{m}} \sum_{i=1}^{m} \underline{\left(x^{(i)} - \mu\right)^2}$$

Density estimation

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

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

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

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

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

$$p(x)$$

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

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

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

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

Anomaly detection algorithm

1. Choose features x_i that you think might be indicative of anomalous examples. $\{x^{(1)}, \ldots, x^{(m)}\}$
2. Fit parameters $\mu_1, \ldots, \mu_n, \sigma_1^2, \ldots, \sigma_n^2$

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

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

$\mu_1, \mu_2, \ldots, \mu_n$

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

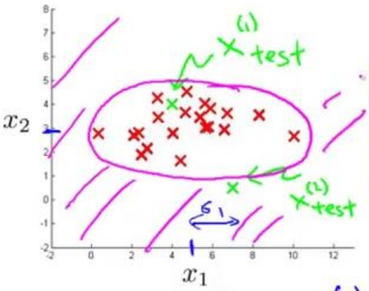
$$\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}(x_j^{(i)} - \mu_j)^2$$

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 $p(x) < \varepsilon$

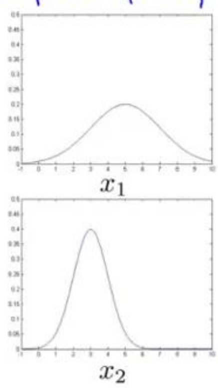
Anomaly detection example



σ_1^2, σ_2^2
$= 4$

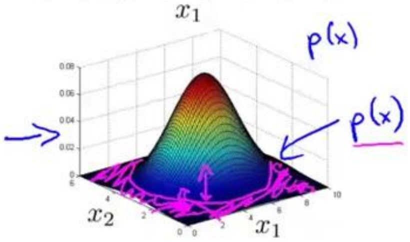
$\mu_1 = 5, \sigma_1 = 2$
$\mu_2 = 3, \sigma_2 = 1$

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



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

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



$\varepsilon = 0.02$

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

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

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: $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ (assume normal examples/not anomalous)
- Cross validation set: $(x_{cv}^{(1)}, y_{cv}^{(1)}), \ldots, (x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})})$
- Test set: $(x_{test}^{(1)}, y_{test}^{(1)}), \ldots, (x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$

Aircraft engines motivating example

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

$\mu_1, \sigma_1^2, \ldots, \mu_n, \sigma_n^2$

- Training set: $\boxed{6000}$ good engines ($y=0$) $p(x) = p(x_1; \mu_1, \sigma_1^2) \ldots p(x_n; \mu_n, \sigma_n^2)$

CV: $\boxed{2000}$ good engines ($y = 0$), $\boxed{10}$ anomalous ($y = 1$)

Test: $\boxed{2000}$ good engines ($y = 0$), $\boxed{10}$ anomalous ($y = 1$)

Alternative:

Training set: $\boxed{6000}$ good engines

- CV: $\boxed{4000}$ good engines ($y = 0$), $\boxed{10}$ anomalous ($y = 1$)
- Test: $\boxed{4000}$ good engines ($y = 0$), $\boxed{10}$ anomalous ($y = 1$)

Algorithm evaluation

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

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

$y=0$

Possible evaluation metrics:

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

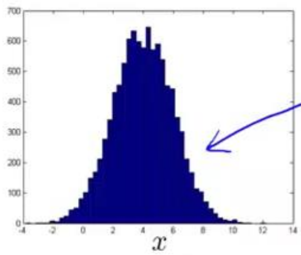
CV

Can also use cross validation set to choose parameter ε

Anomaly detection	vs. Supervised learning
→ Very small number of positive examples ($\underline{y = 1}$). ($\underline{0\text{-}20}$ is common). → Large number of negative ($\underline{y = 0}$) examples. $\boxed{p(x)}$ ←	Large number of positive and negative 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.	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	vs. Supervised learning
→ • Fraud detection $y=1$ →	• Email spam classification ←
→ • Manufacturing (e.g. aircraft engines)	• Weather prediction (sunny/rainy/etc). ←
→ • Monitoring machines in a data center	• Cancer classification ←

Non-gaussian features



hist

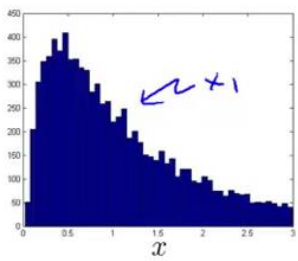
$p(x_i; \mu_i, \sigma_i^2)$

$x_1 \leftarrow \log(x_1)$

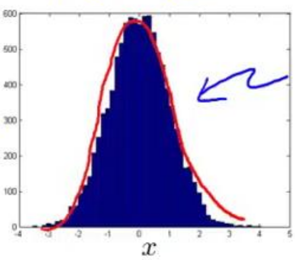
$x_2 \leftarrow \log(x_2+1)$ $\quad \log(x_2 + c)$

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

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



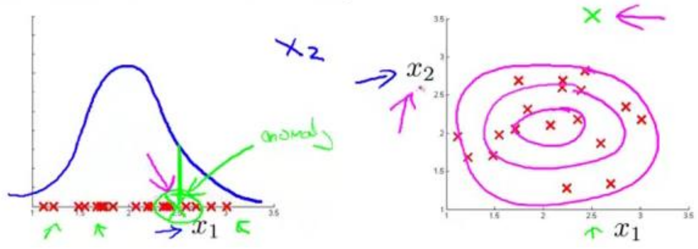
$\xrightarrow{\log(x)}$



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

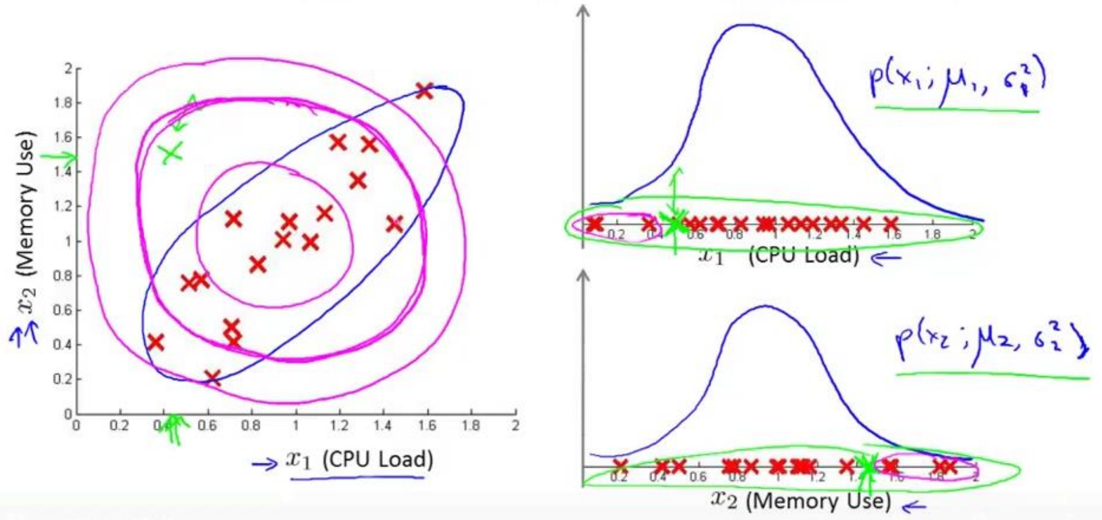
Choose features that might take on unusually large or small values in the event of an anomaly.

- x_1 = memory use of computer
- x_2 = number of disk accesses/sec
- x_3 = CPU load
- x_4 = network traffic

$$x_5 = \frac{\text{CPU load}}{\text{network traffic}} \qquad x_6 = \frac{(\text{CPU load})^2}{\text{network traffic}}$$

异常检测算法的延伸　　应用 Lee 多元高斯分布

Motivating example: Monitoring machines in a data center



Multivariate Gaussian (Normal) distribution

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

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

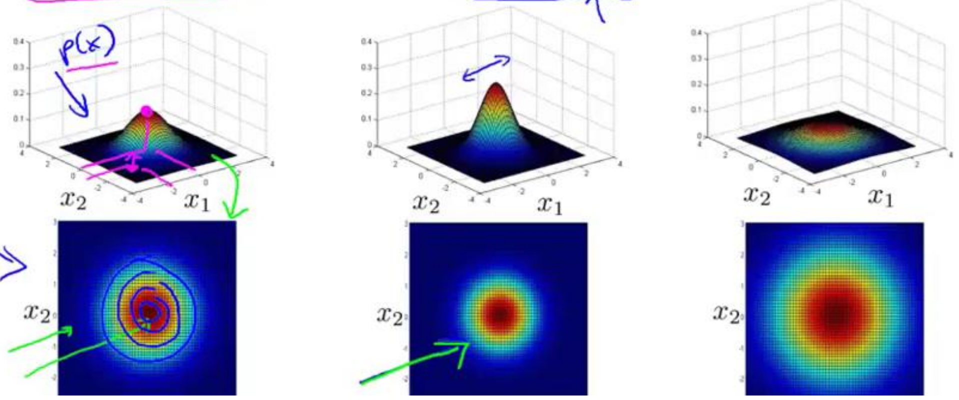
Parameters: $\mu \in \mathbb{R}^n$, $\boxed{\Sigma \in \mathbb{R}^{n \times n}}$ (covariance matrix)

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

$|\Sigma|$ = determinant of Σ | det(Sigma)

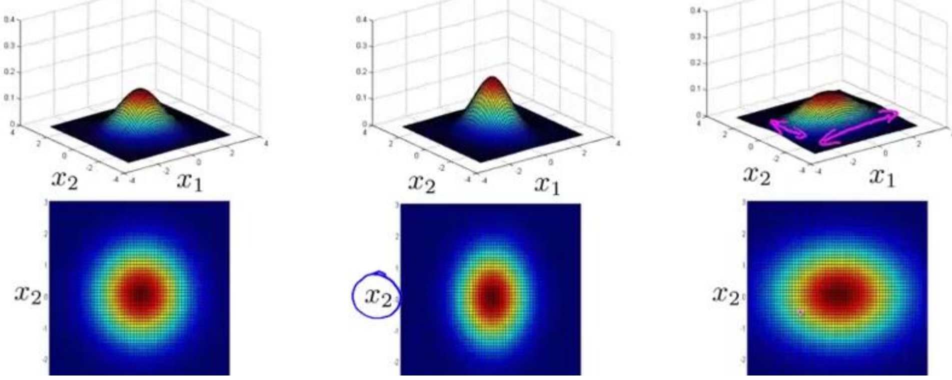
Multivariate Gaussian (Normal) examples

$\boxed{\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \; \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}$ $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \; \boxed{\Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix}}$ $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \; \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$



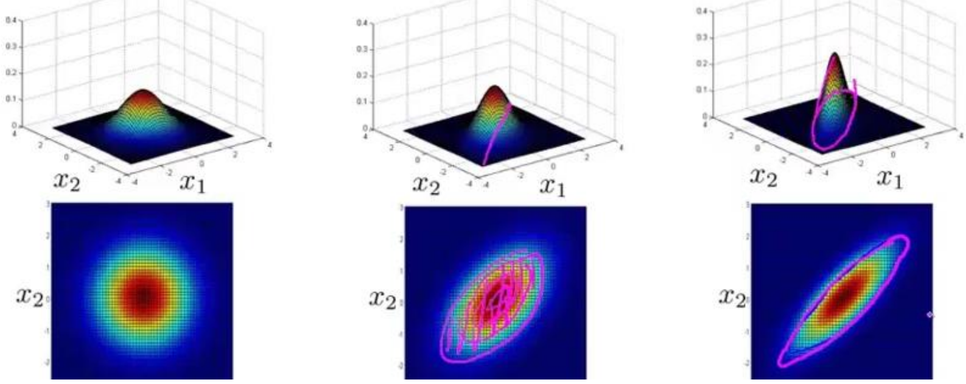
Multivariate Gaussian (Normal) examples

$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $\quad$ $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 1 \end{bmatrix}$ $\quad$ $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$



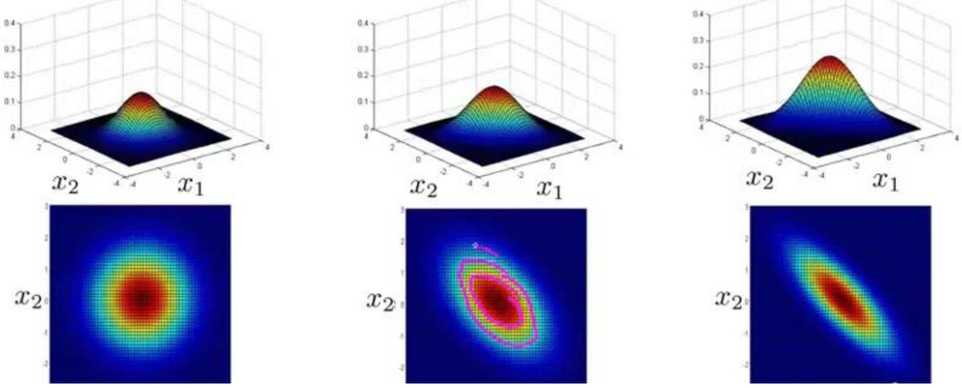
Multivariate Gaussian (Normal) examples

$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $\quad$ $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$ $\quad$ $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$



Multivariate Gaussian (Normal) examples

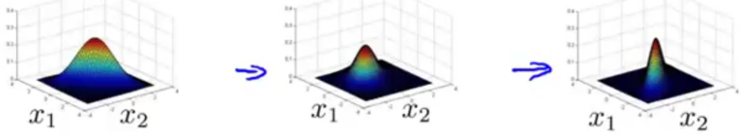
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $\quad$ $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$ $\quad$ $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}$



Multivariate Gaussian (Normal) distribution

Parameters μ, Σ $\quad \mu \in \mathbb{R}^n \quad \Sigma \in \mathbb{R}^{n \times n}$

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



Parameter fitting:

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

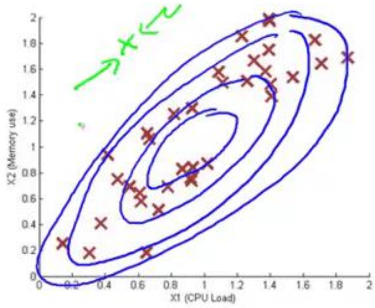
$$\mu = \frac{1}{m} \sum_{i=1}^{m} x^{(i)} \qquad \Sigma = \frac{1}{m} \sum_{i=1}^{m} (x^{(i)} - \mu)(x^{(i)} - \mu)^T$$

Anomaly detection with the multivariate Gaussian

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

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

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



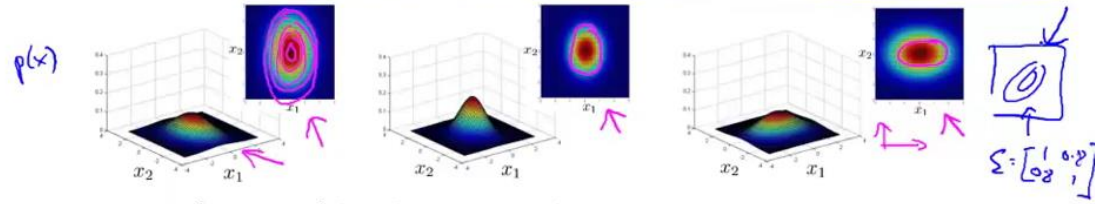
2. Given a new example x, compute

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

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

Relationship to original model

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



Corresponds to multivariate Gaussian

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

where $\Sigma = \begin{bmatrix} \sigma_1^2 & & 0 \\ & \sigma_2^2 & \\ & \ddots & \\ 0 & & \sigma_n^2 \end{bmatrix}$

→ Original model vs. → Multivariate Gaussian

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. $\rightarrow \boxed{x_3 = \frac{x_1}{x_2} = \frac{\text{CPU load}}{\text{memory}}}$

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

OK even if m (training set size) is small

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

$\Sigma \in \mathbb{R}^{n \times n}$ $\quad \Sigma^{-1}$

Computationally more expensive

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

$x_1 = \cancel{x_2}$
$\cancel{x_3} = x_4 + x_5$

Must have $\boxed{m > n}$ or else Σ is non-invertible. $\rightarrow \boxed{m \geq 10n}$