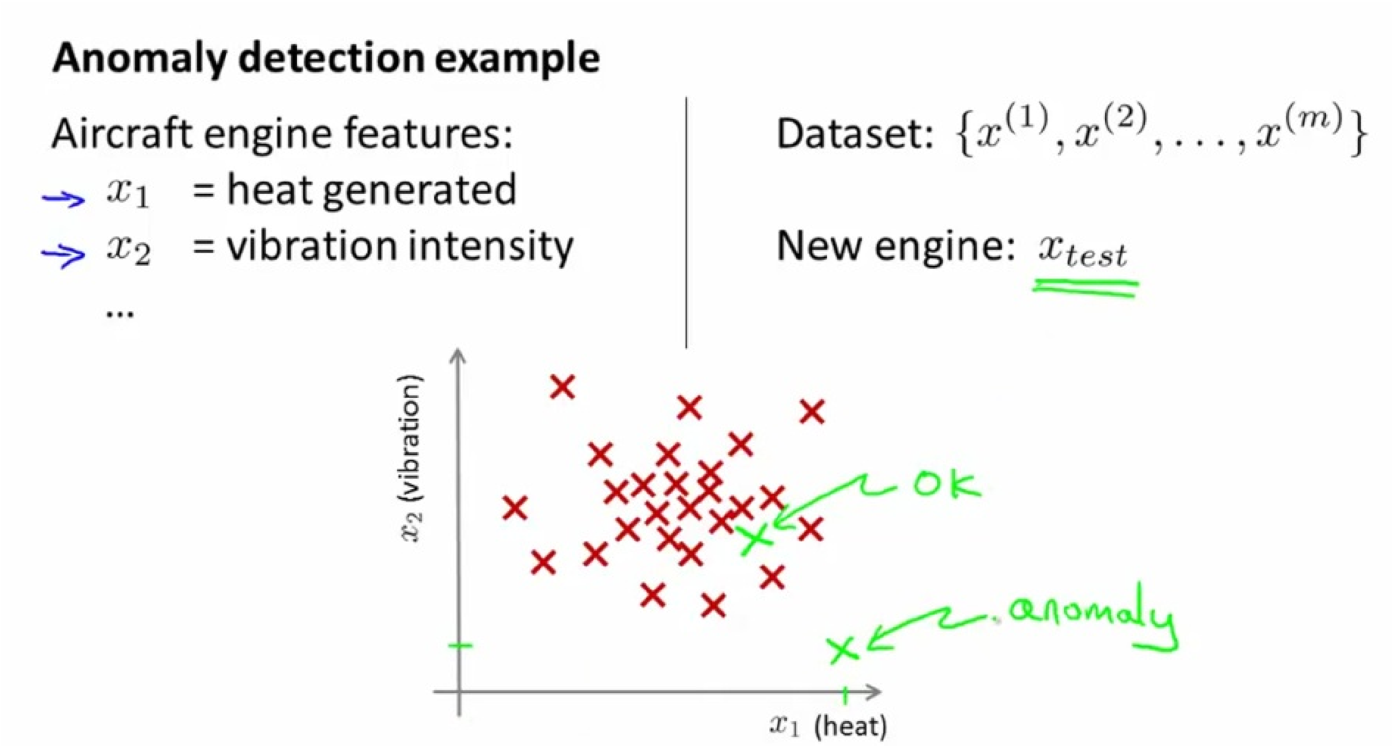
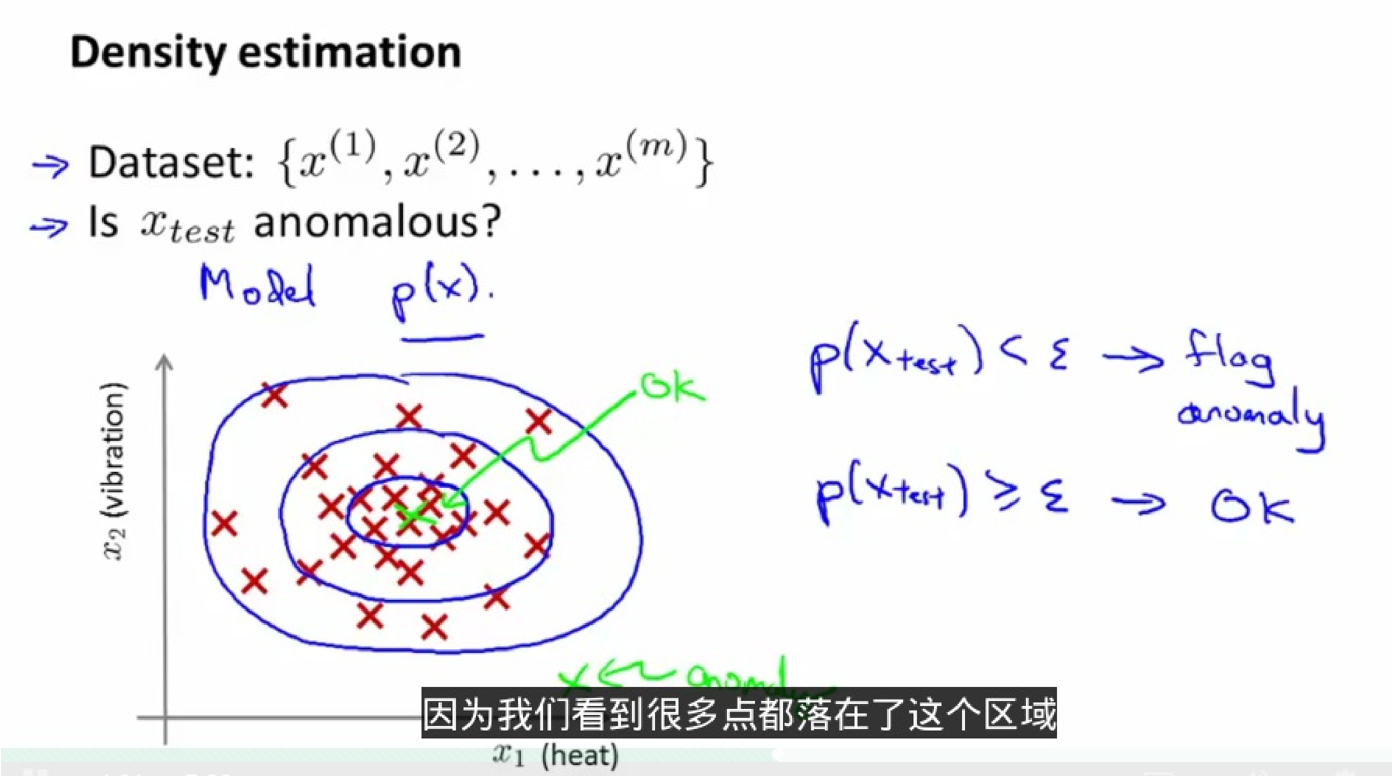
Anomaly Detection：

异常检测的例子：





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 Distribution）：

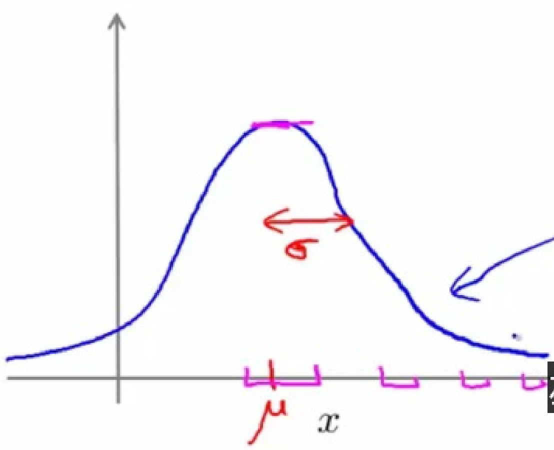
Gaussian (Normal) distribution

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

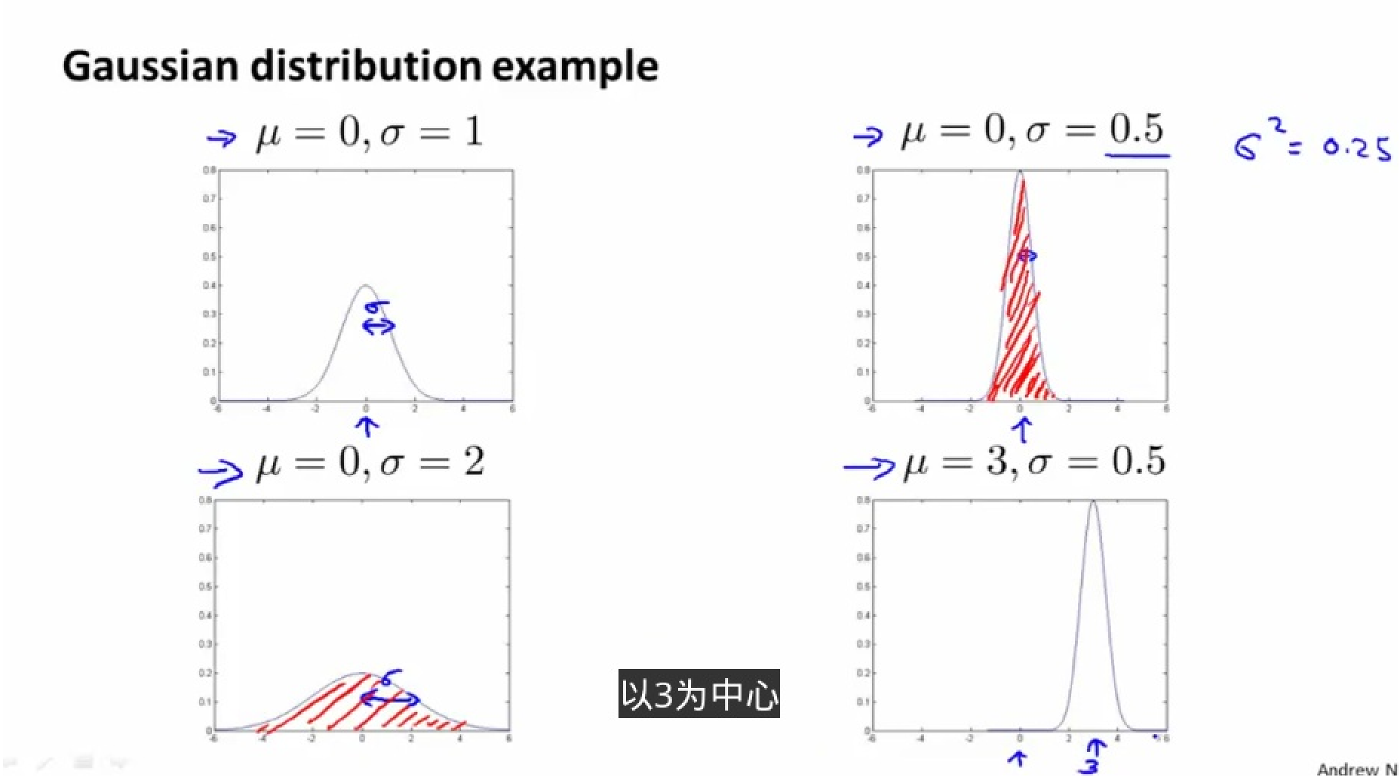
$$x \sim \mathcal{N}(\mu, \sigma^2)$$

"distributed as"

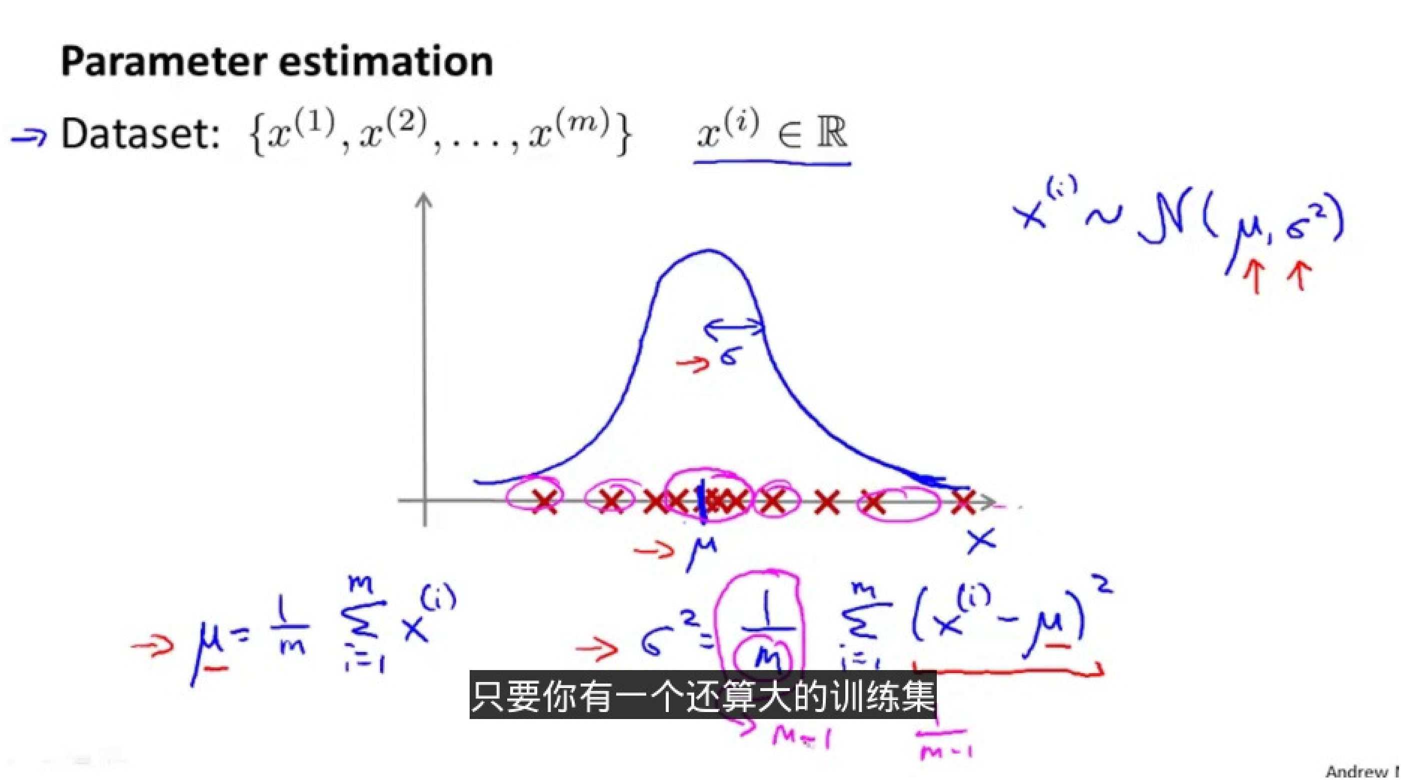
$$p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$



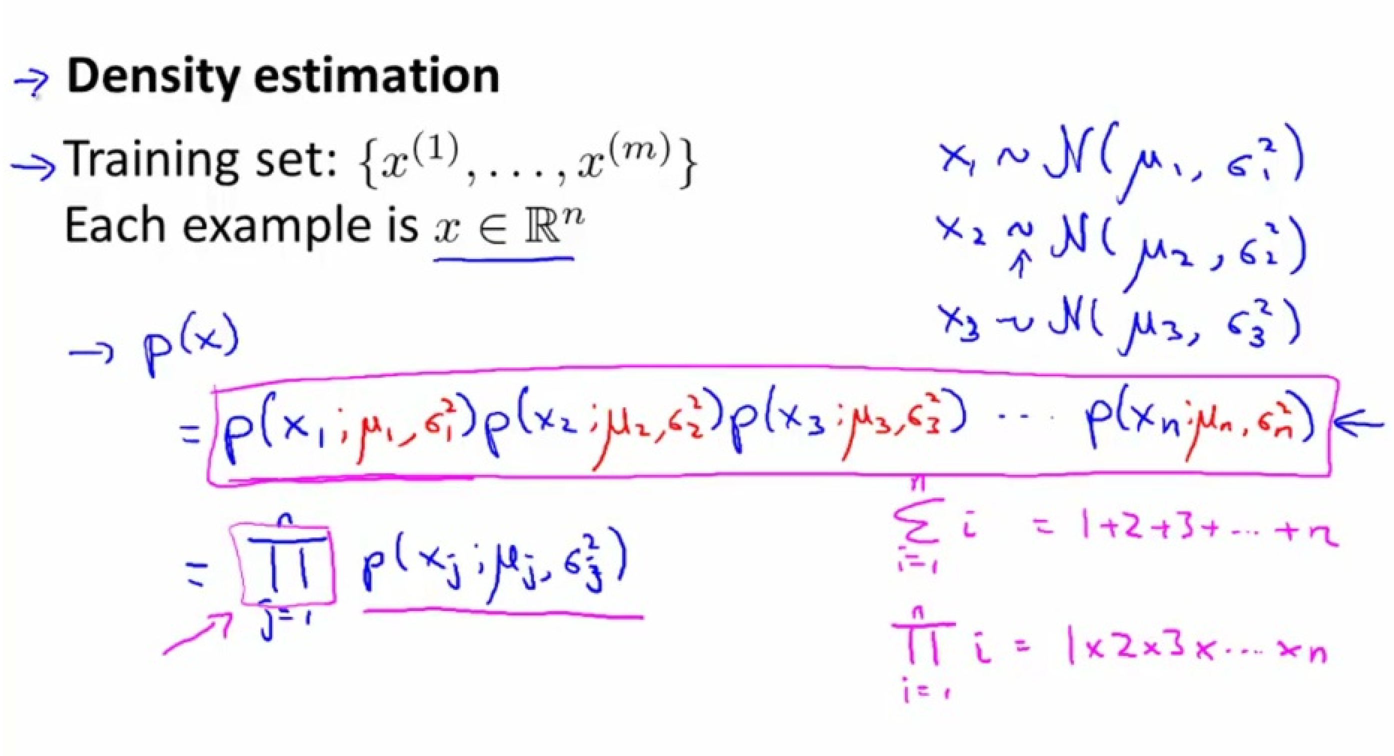
左边这幅图



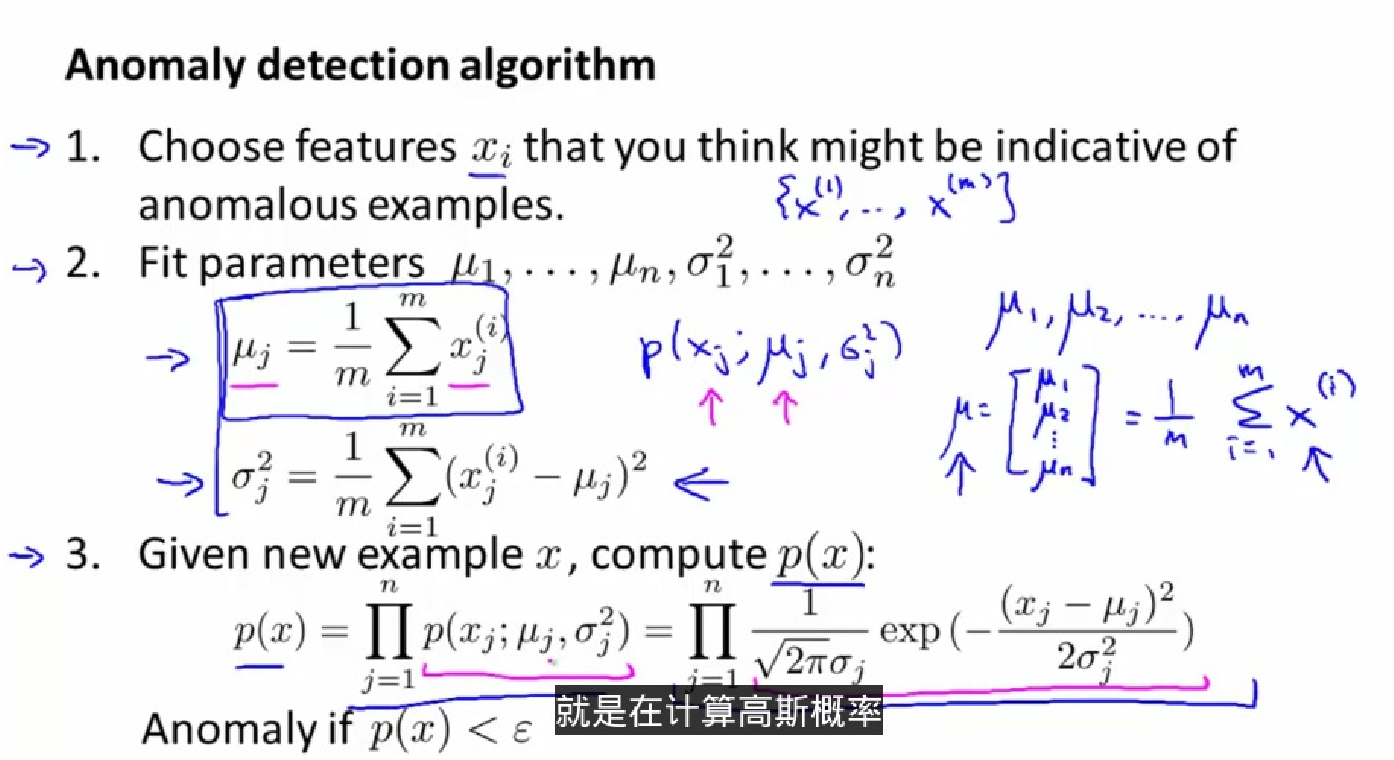
参数估计问题：给定数据集，从而估计出高斯分布参数mu和sigma。



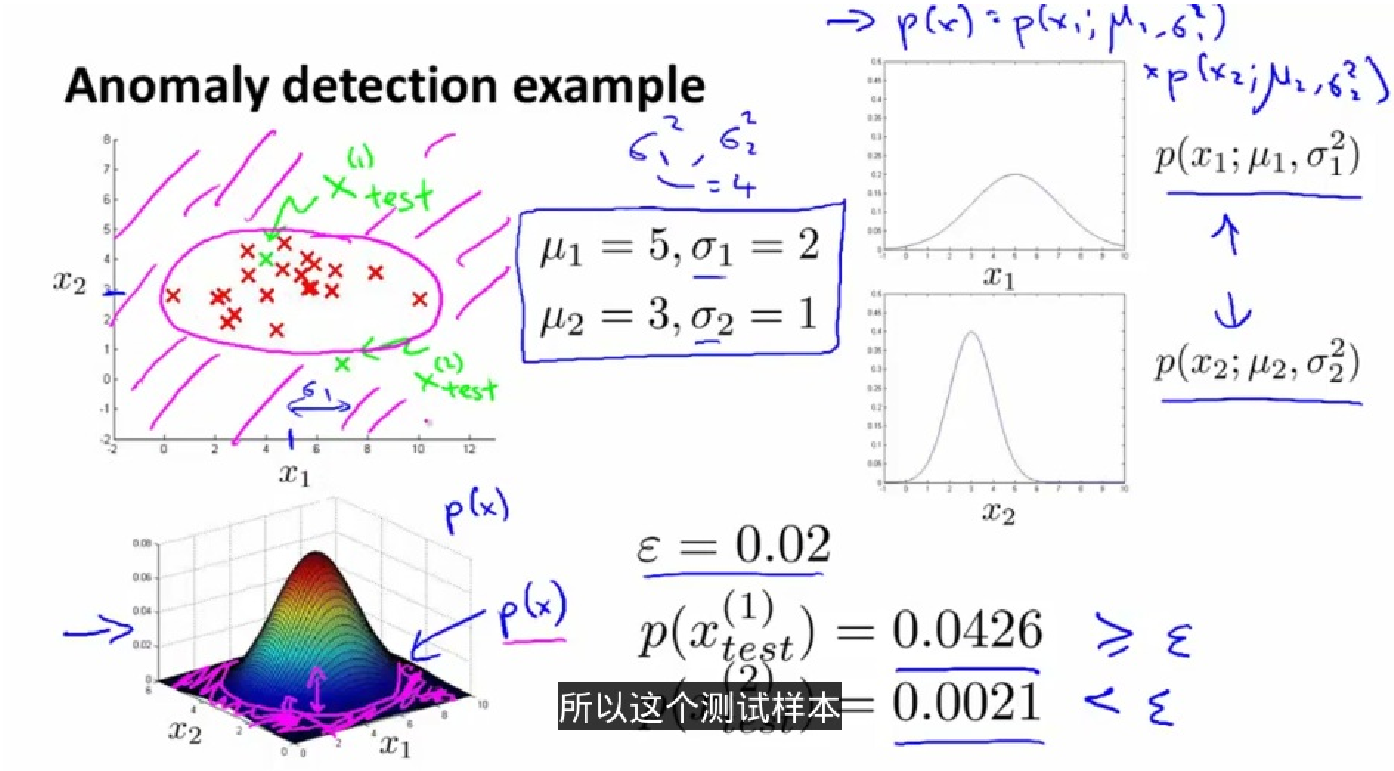
异常检测算法，密度估计


Density estimation
Training set: $\{x^{(1)}, \ldots, x^{(m)}\}$
Each example is $x \in \mathbb{R}^n$
$x_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$
$x_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$
$x_3 \sim \mathcal{N}(\mu_3, \sigma_3^2)$
$p(x)$
$= p(x_1; \mu_1, \sigma_1^2) p(x_2; \mu_2, \sigma_2^2) p(x_3; \mu_3, \sigma_3^2) \cdots p(x_n; \mu_n, \sigma_n^2)$
$= \prod_{j=1}^{n} p(x_j; \mu_j, \sigma_j^2)$
$\sum_{i=1}^{n} i = 1+2+3+\cdots+n$
$\prod_{i=1}^{n} i = 1 \times 2 \times 3 \times \cdots \times n$

异常检测算法：


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 $\mu_1, \ldots, \mu_n, \sigma_1^2, \ldots, \sigma_n^2$
$\mu_j = \frac{1}{m} \sum_{i=1}^{m} x_j^{(i)}$
$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^{m} (x_j^{(i)} - \mu_j)^2$
$p(x_j; \mu_j, \sigma_j^2)$
$\mu_1, \mu_2, \ldots, \mu_n$
$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix} = \frac{1}{m} \sum_{i=1}^{m} x^{(i)}$
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 $p(x) < \varepsilon$
就是在计算高斯概率

举个栗子：



开发一个异常检测系统：

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 $(y=0)$ $p(x) = p(x_1; \mu_1, \sigma_1^2) \cdots p(x_n; \mu_n, \sigma_n^2)$
- CV: $\boxed{2000}$ good engines ($y = 0$), $\boxed{10}$ anomalous ($y = 1$)
- Test: $\boxed{2000}$ good engines ($y = 0$), $\boxed{10}$ anomalous ($y = 1$)

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:

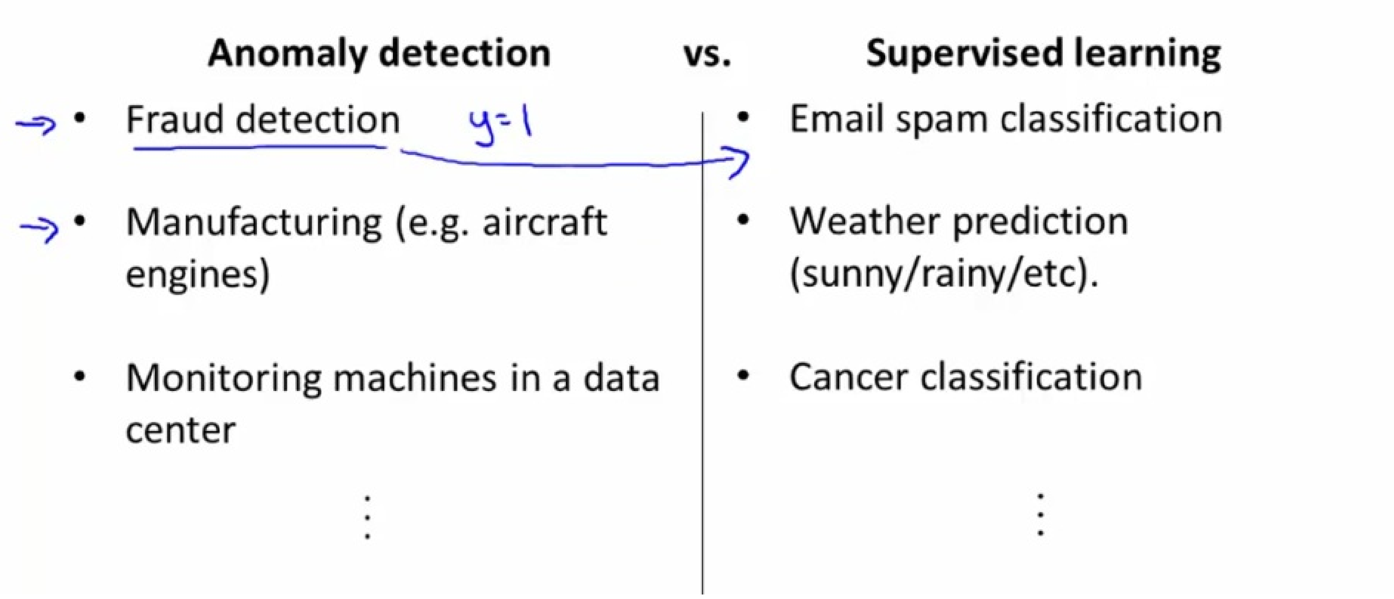
- True positive, false positive, false negative, true negative
- Precision/Recall
- F_1-score

Can also use cross validation set to choose parameter ε

但是如果有了一些带标签的数据，那么为什么不用监督学习的方法来进行异常检测呢？

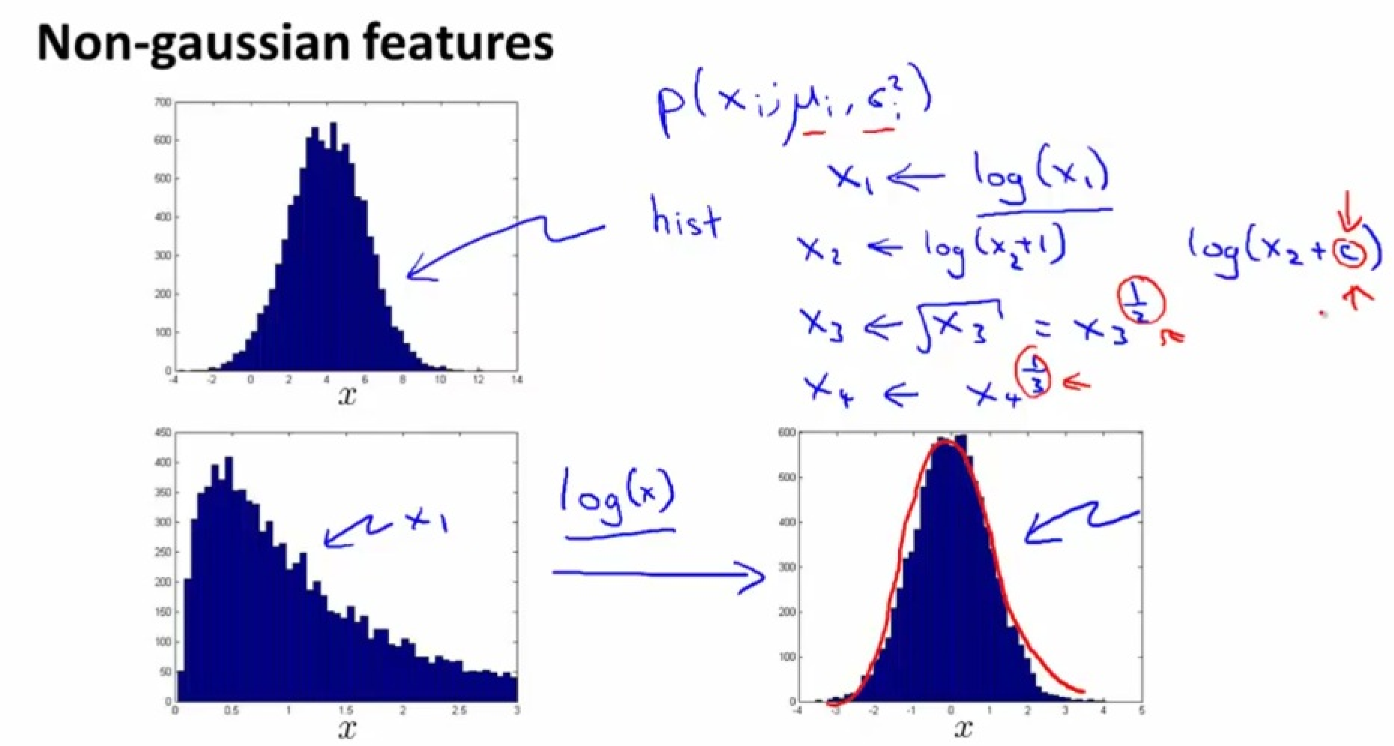
Anomaly detection	vs. Supervised learning
Very small number of positive examples ($y = 1$). (0-20 is common).	Large number of positive and negative examples.
Large number of negative ($y = 0$) examples. $\boxed{p(x)}$	
Many different "types" of anomalies. Hard for any algorithm to learn from positive examples what the anomalies look like; future anomalies may look nothing like any of the anomalous examples we've seen so far.	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.

Andrew Ng



异常检测中的特征选择：

可以对参数进行调整从而让他看起来更像是高斯分布：



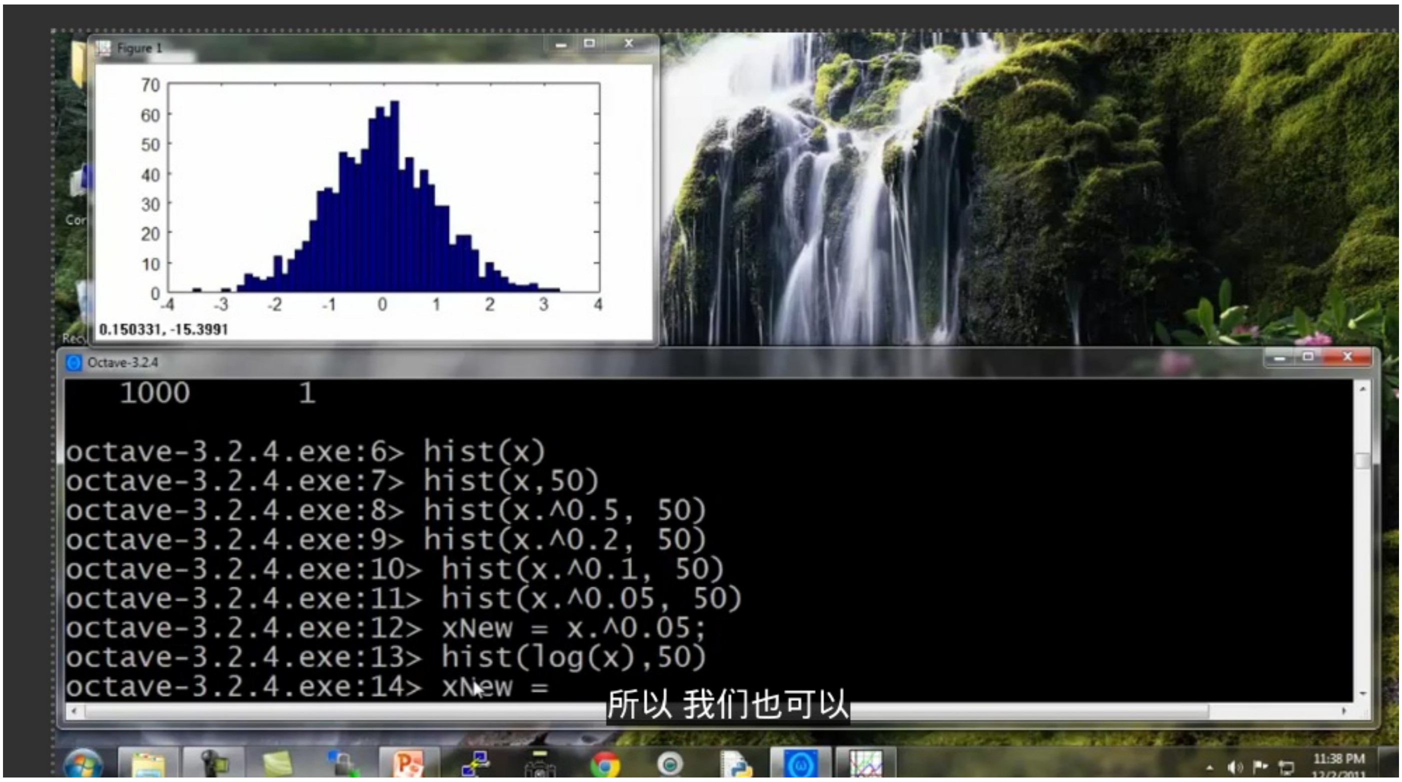
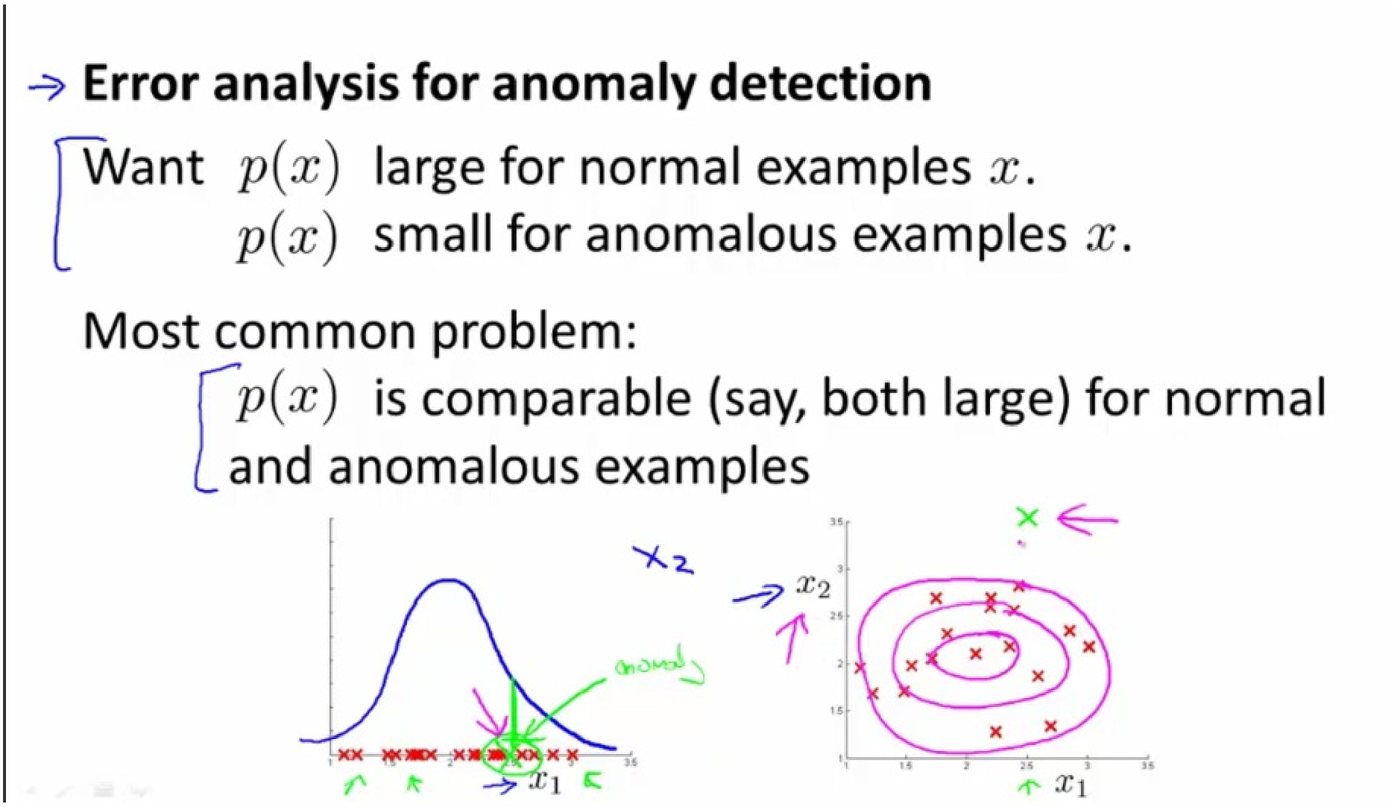


Figure 1
0.150331, -15.3991
Octave-3.2.4
1000 1
octave-3.2.4.exe:6> hist(x)
octave-3.2.4.exe:7> hist(x,50)
octave-3.2.4.exe:8> hist(x.^0.5, 50)
octave-3.2.4.exe:9> hist(x.^0.2, 50)
octave-3.2.4.exe:10> hist(x.^0.1, 50)
octave-3.2.4.exe:11> hist(x.^0.05, 50)
octave-3.2.4.exe:12> xNew = x.^0.05;
octave-3.2.4.exe:13> hist(log(x),50)
octave-3.2.4.exe:14> xNew =
所以 我们也可以

异常检测的误差分析：



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
x_2
anomaly
x_1
x_2
x_1

Monitoring computers in a data center

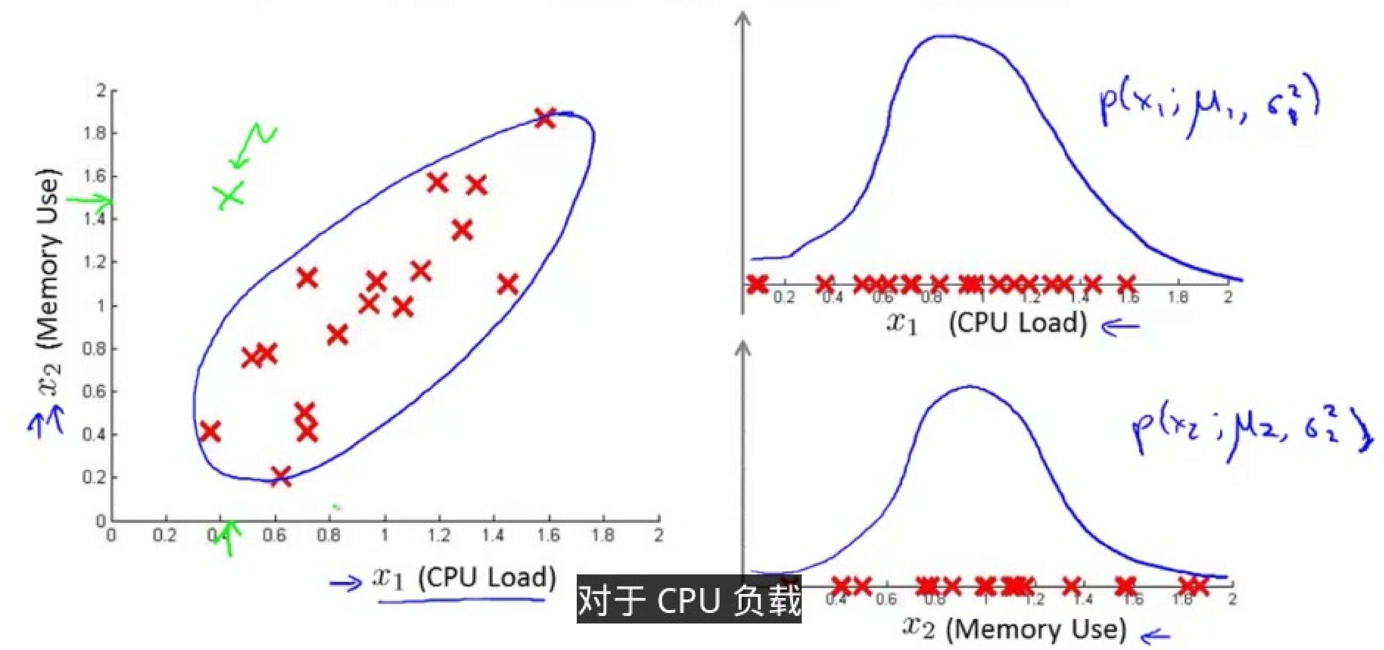
Choose features that might take on unusually large or small values in the event of an anomaly.

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

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

多元高斯分布（Multivariate Gaussian Distribution）：

Motivating example: Monitoring machines in a data center



Multivariate Gaussian (Normal) distribution

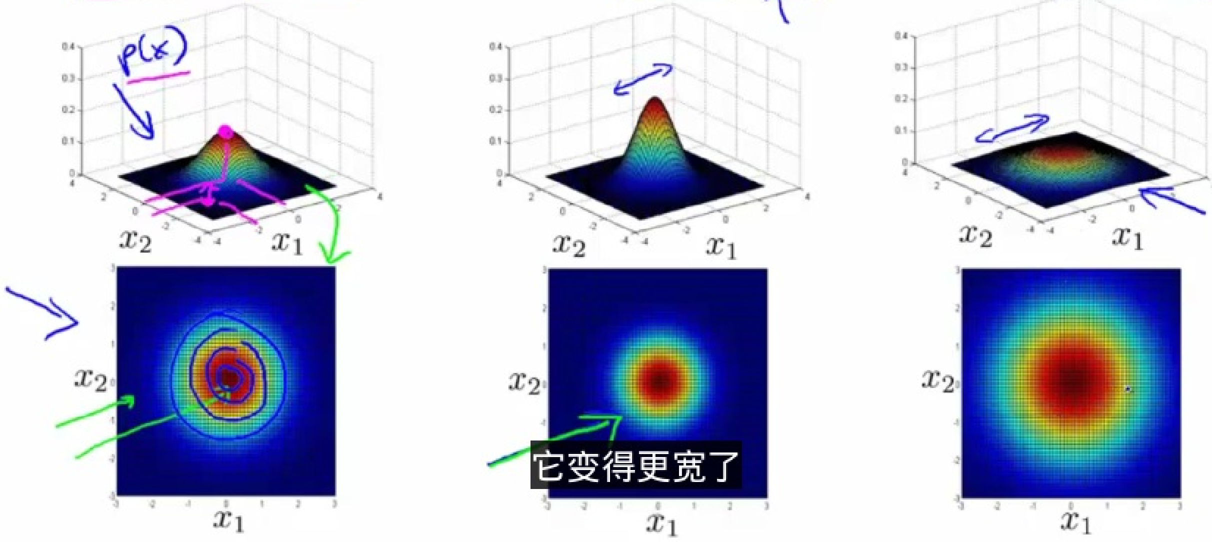
$x \in \mathbb{R}^n$. Don't model $p(x_1), p(x_2), \ldots,$ etc. separately.
Model $p(x)$ all in one go.
Parameters: $\mu \in \mathbb{R}^n$, $\boxed{\Sigma \in \mathbb{R}^{n \times n}}$ (covariance matrix)

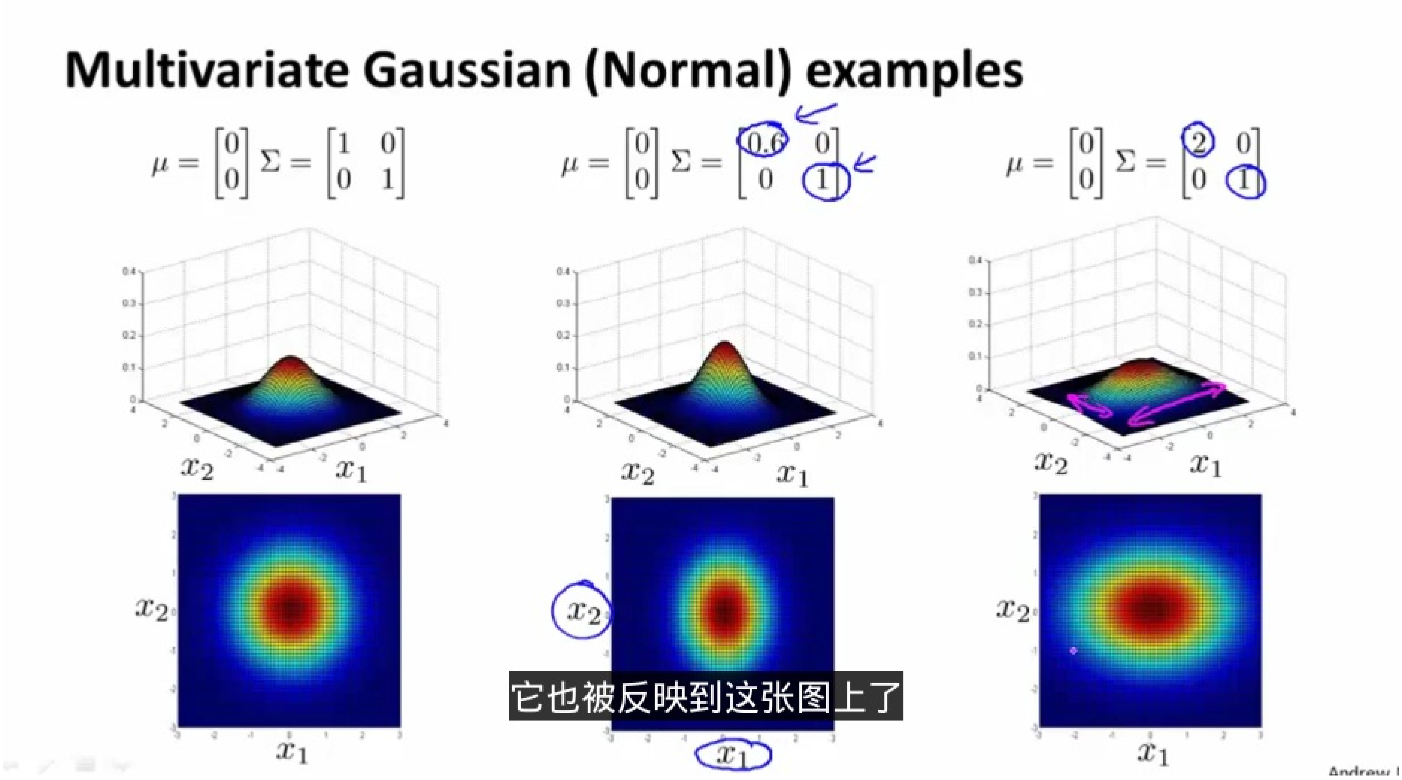
$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

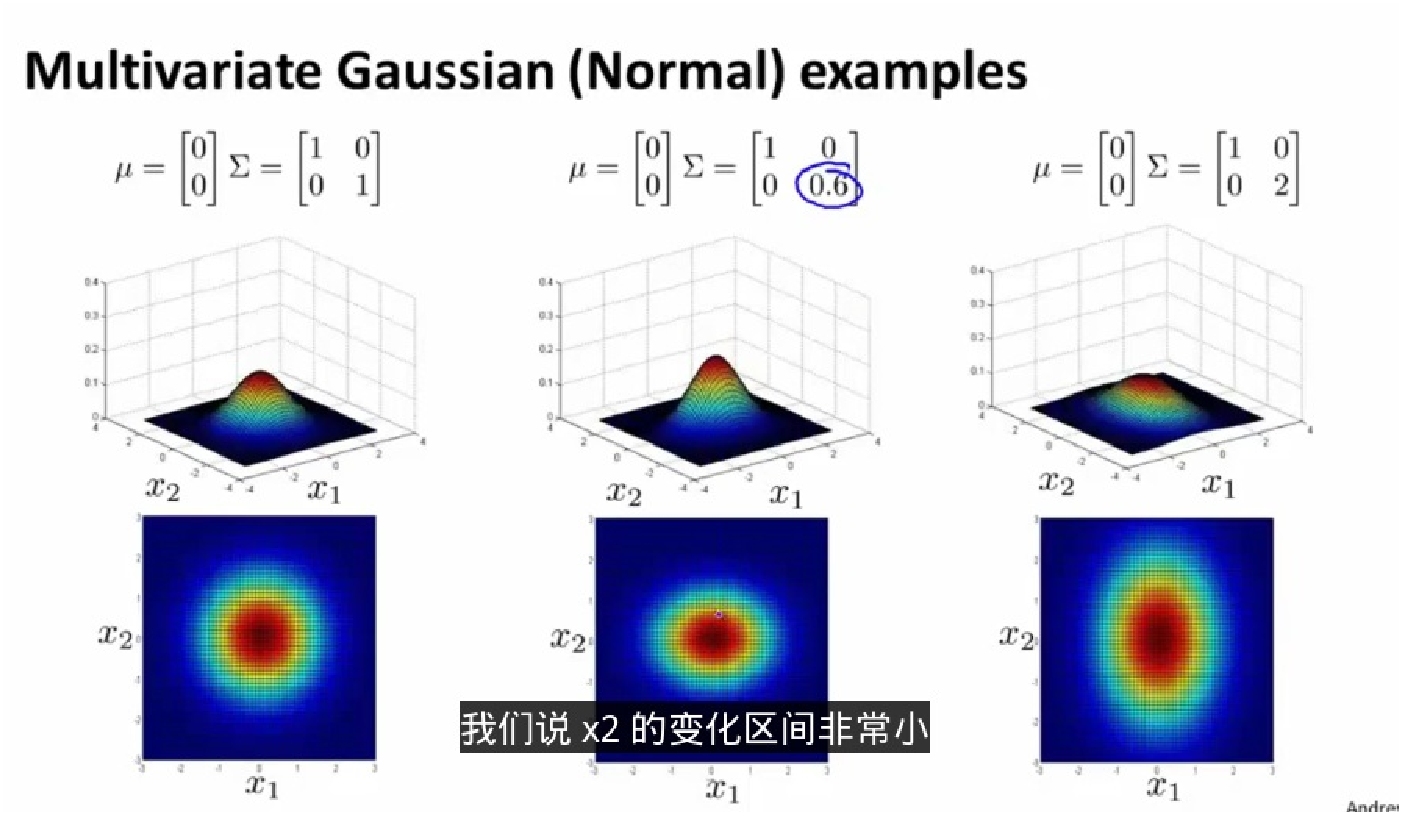
$|\Sigma|$ = determinant of Σ | det(Sigma)

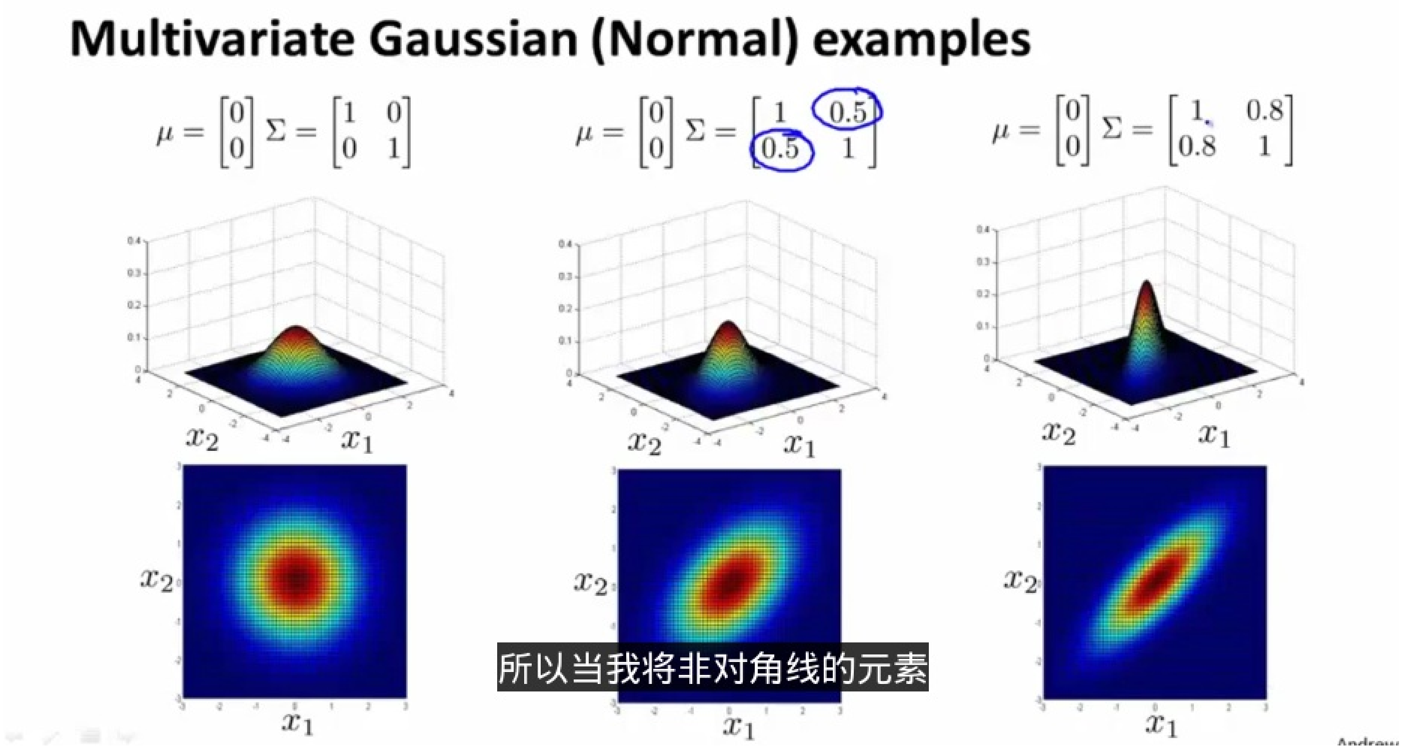
Multivariate Gaussian (Normal) examples

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

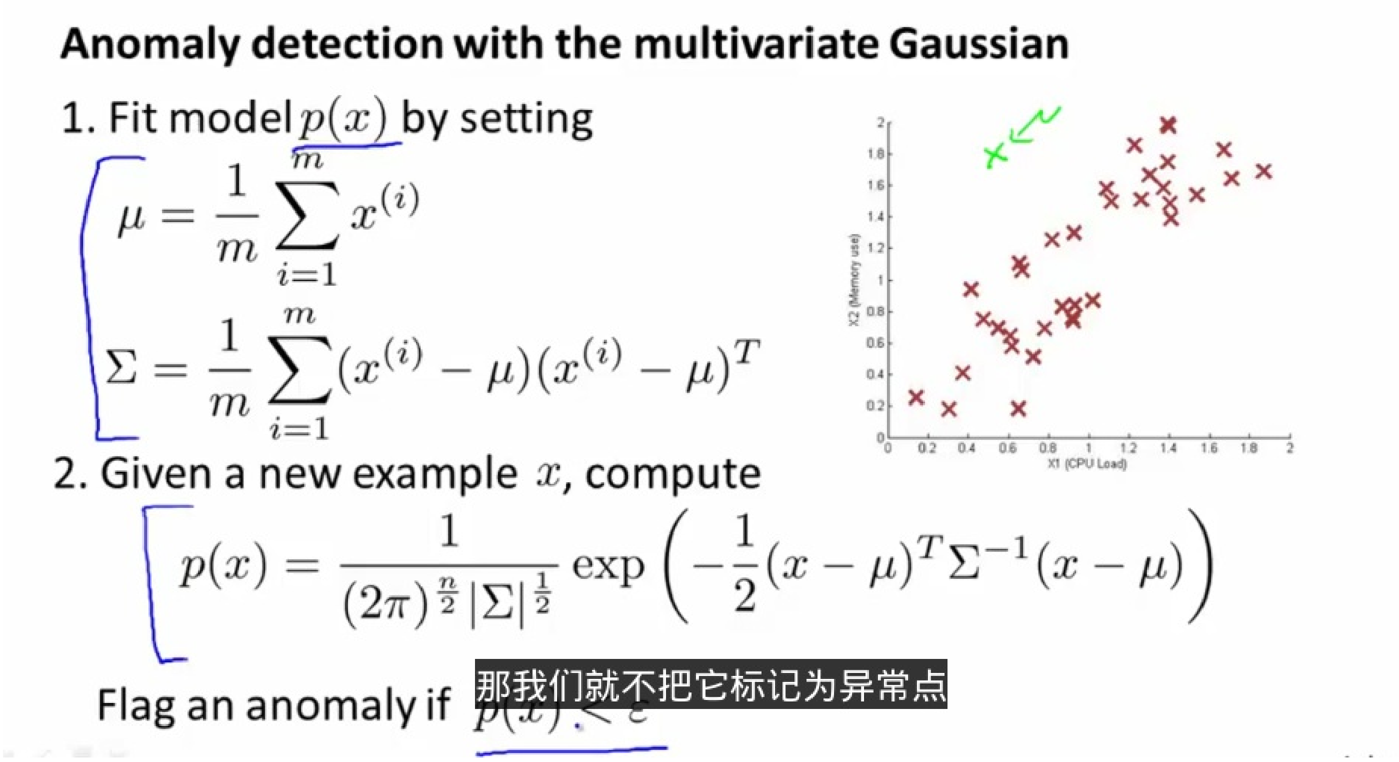



Multivariate Gaussian (Normal) examples
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 1 \end{bmatrix}$
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$
x_2
x_1
它也被反映到这张图上了


Multivariate Gaussian (Normal) examples
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 0.6 \end{bmatrix}$
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$
x_2
x_1
我们说 x2 的变化区间非常小


Multivariate Gaussian (Normal) examples
$\mu = \begin{bmatrix}0\\0\end{bmatrix} \Sigma = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$
$\mu = \begin{bmatrix}0\\0\end{bmatrix} \Sigma = \begin{bmatrix}1 & 0.5\\0.5 & 1\end{bmatrix}$
$\mu = \begin{bmatrix}0\\0\end{bmatrix} \Sigma = \begin{bmatrix}1 & 0.8\\0.8 & 1\end{bmatrix}$
x_2 x_1
所以当我将非对角线的元素

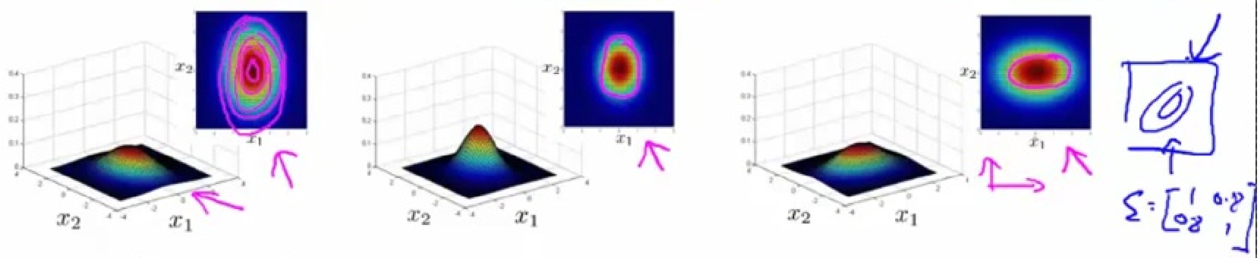
多元高斯分布异常检测：


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 $p(x) < \varepsilon$
X2 (Memory use)
X1 (CPU Load)
那我们就不把它标记为异常点

Relationship to original model

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

$p(x)$



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

所以你不能给不同特征变量之间的相关性建模

Andrew Ng

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.

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

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

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

$x_1 = x_2$

$x_3 = x_4 + x_5$

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

$m \geq 10n$