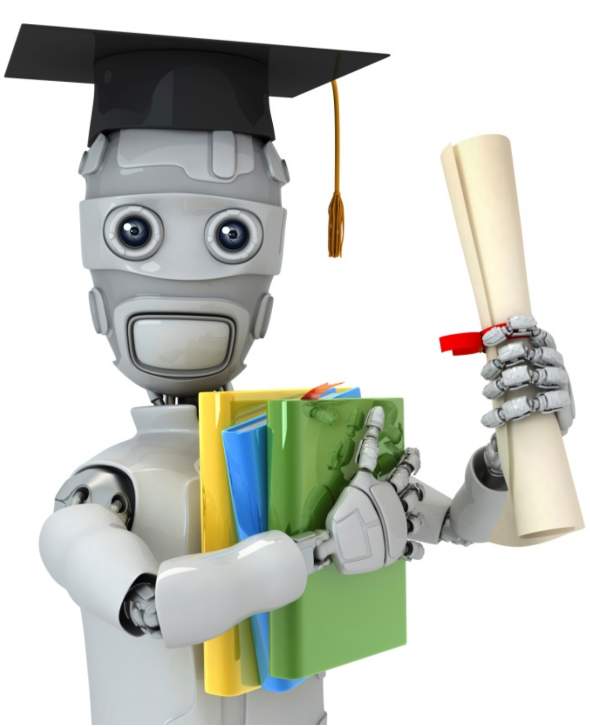


Machine Learning

Anomaly detection

Problem motivation

Anomaly detection example

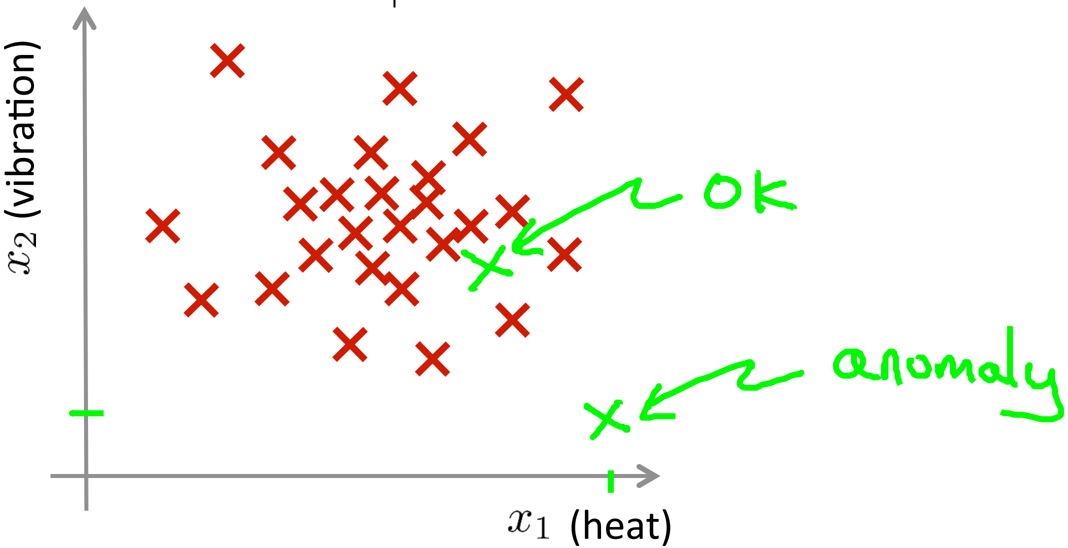
Aircraft engine features:

- x_1 = heat generated
- x_2 = vibration intensity

…

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

New engine: x_{test}

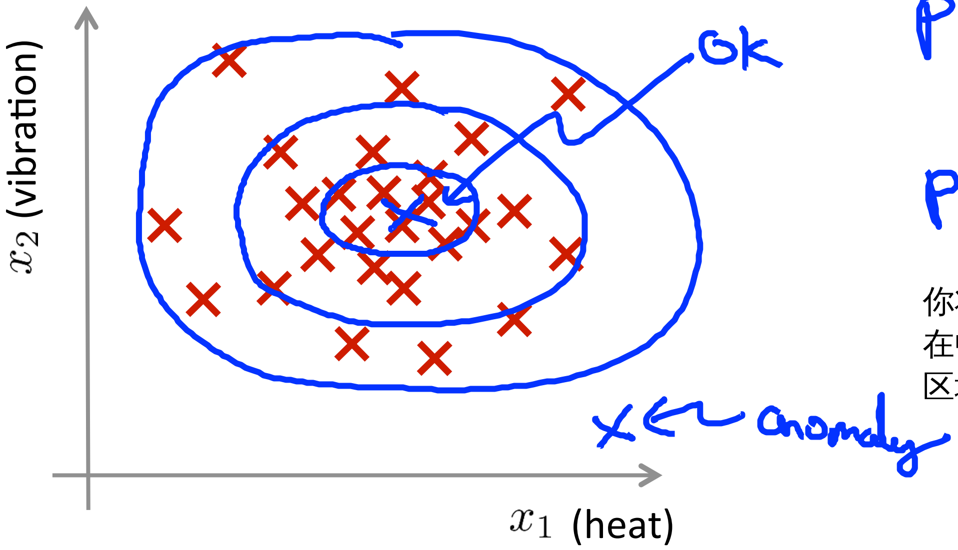


Density estimation

→ 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

你将很可能发现飞机引擎 很可能发现模型p(x) 将会认为在中心区域的这些点 有很大的概率值 而稍微远离中心区域的点概率会小一些

异常检测算法有如下应用案例

也许x1是用户登陆的频率 x2也许是 用户访问 某个页面的次数 或者交易次数 也许x3是 用户在论坛上发贴的次数 x4是 用户的 打字速度

Anomaly detection example

也许异常检测 最常见的应用是 是欺诈检测

- 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

$p(x)$

异常检测的另一个例子是在工业生产领域 事实上 我们之前已经谈到过 飞机引擎的问题

- Manufacturing

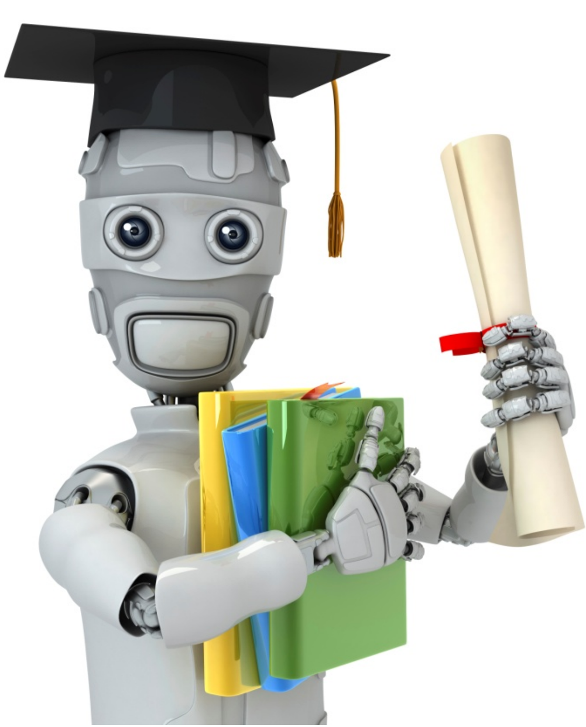
第三个应用是 数据中心的计算机监控

- 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.
  - ...

$p(x) < \varepsilon$

Anomaly detection

Gaussian distribution



Machine Learning

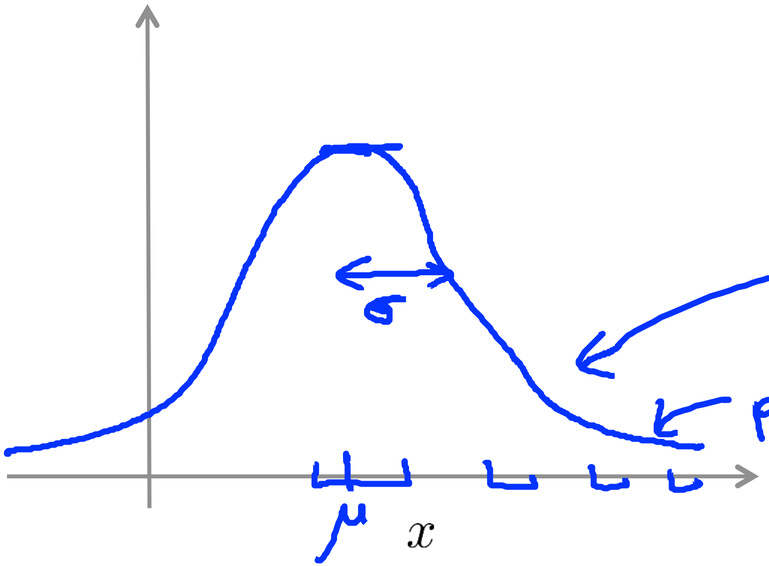
Gaussian (Normal) distribution

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

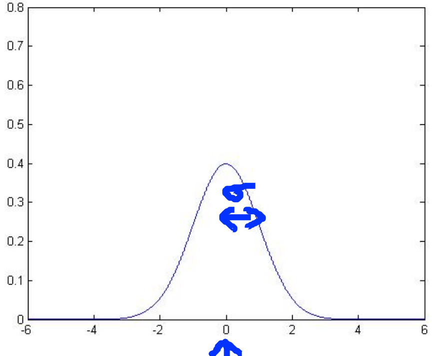


$p(x;\mu,\sigma^2)$

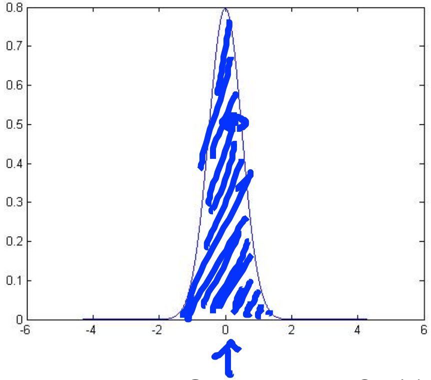
$$p(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Gaussian distribution example

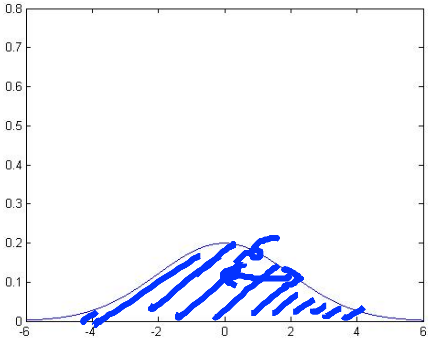
$\mu = 0, \sigma = 1$



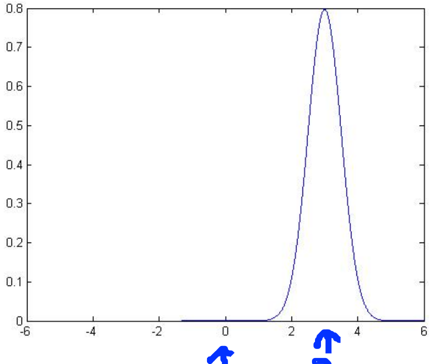
$\mu = 0, \sigma = \underline{0.5}$ $\sigma^2 = 0.25$



$\mu = 0, \sigma = 2$



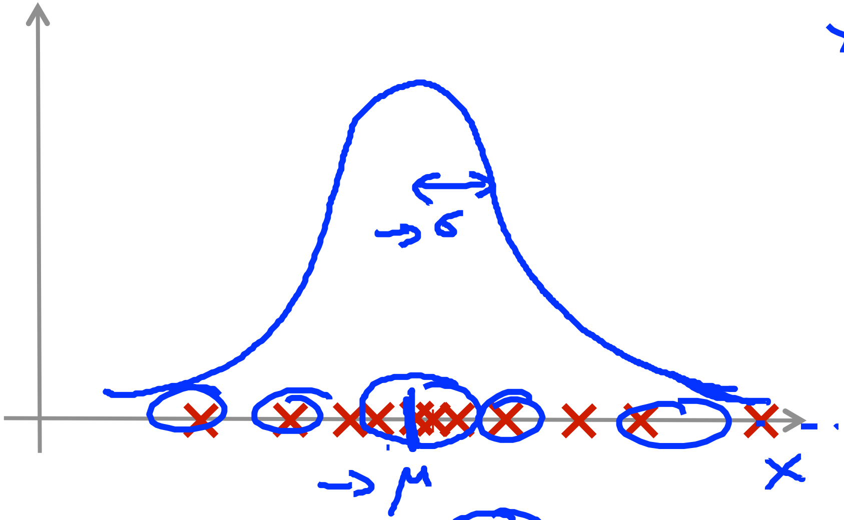
$\mu = 3, \sigma = 0.5$



Parameter estimation

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

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



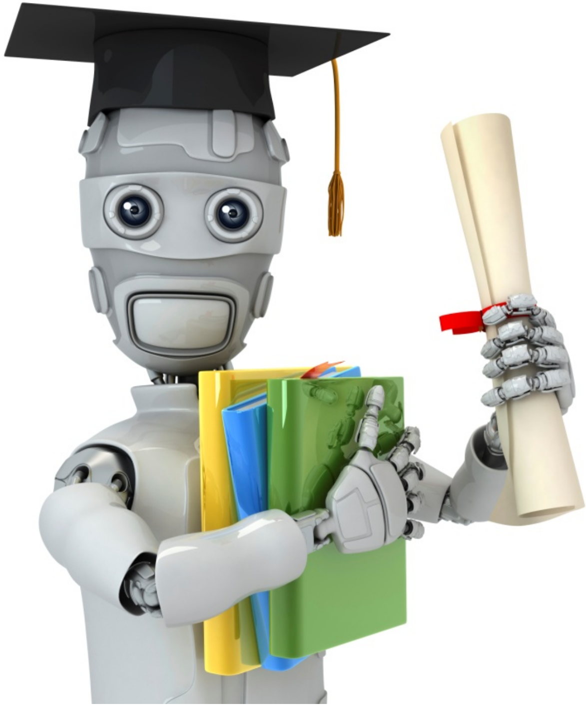
$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$$

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

$m-1$ $\qquad \frac{1}{m-1}$

在实际使用中 到底是选择使用1/m还是1/(m-1)其实区别很小. 在机器学习领域大部分人更习惯使用1/m这个版本的公式

Andrew Ng



Machine Learning

Anomaly detection

Algorithm

Density estimation

假如说我们有一个无标签的训练集 共有 m 个训练样本

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

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

我们要从数据中 建立一个 p(x) 概率模型

假定x1分布 服从高斯正态分布...
这就是我要说的模型

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

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

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

$p(x)$

即使这个独立的假设不成立 这个算法的效果也还不错

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

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

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

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

Anomaly detection algorithm

這句話很misleading, 可以不看

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

(要看的): 给出一组 $\{x^{(1)}, \ldots, x^{(m)}\}$ m 个无标签数据构成的训练集

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

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

j表示分量

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

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

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

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

当给出一个新样本时, 你想要知道是否出现异常

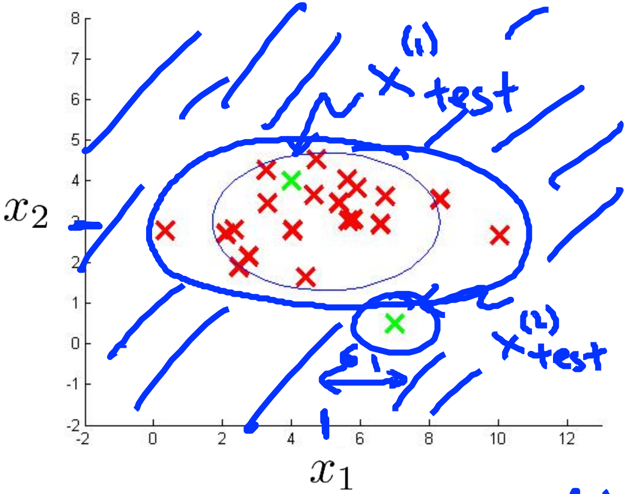
3. Given new example x, compute $p(x)$:

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

Anomaly if $p(x) < \varepsilon$

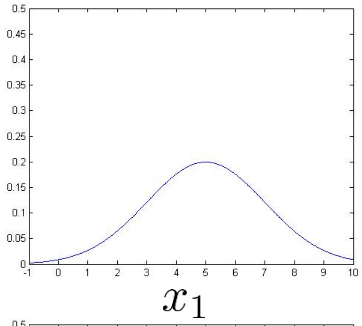
Anomaly detection example

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

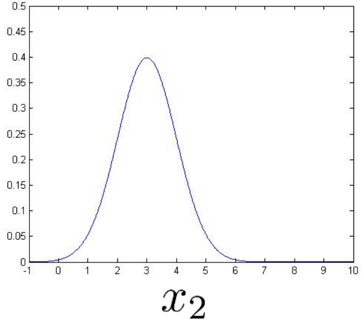


$\sigma_1^2, \sigma_2^2 = 4$

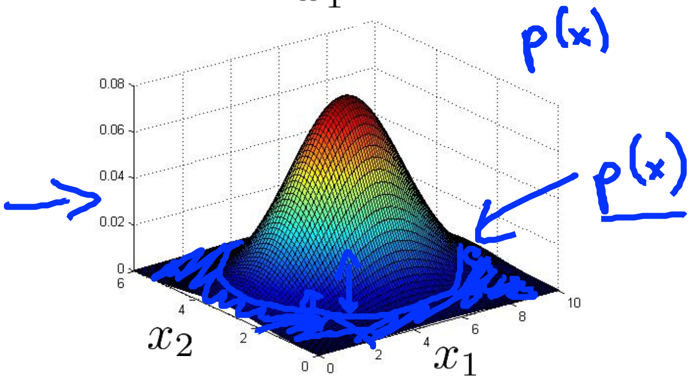
$$\boxed{\begin{aligned} \mu_1 &= 5, \sigma_1 = 2 \\ \mu_2 &= 3, \sigma_2 = 1 \end{aligned}}$$



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



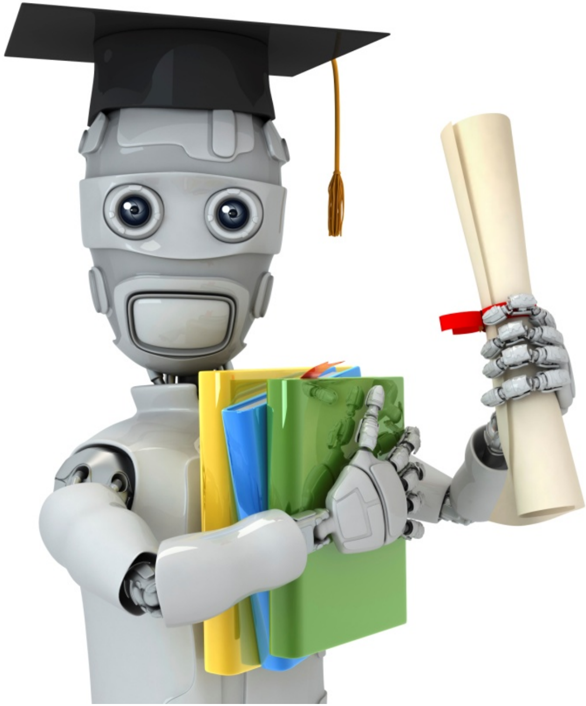
$p(x_2; \mu_2, \sigma_2^2)$



$\varepsilon = 0.02$ 我会在后面讲到如何选取 ε 的值

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

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



Machine Learning

Anomaly detection

Developing and evaluating an anomaly detection system

The importance of real-number evaluation 意思就是評價一個算法好不好

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

y=1

Andrew Ng

Aircraft engines motivating example

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

training set中無anomalous

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

Training set: 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: 2000 good engines ($y = 0$), 10 anomalous ($y = 1$)
Test: 2000 good engines ($y = 0$), 10 anomalous ($y = 1$)

Alternative: 其实我真的不推荐这么分 但就有人喜欢这么分
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 $\{x^{(1)}, \ldots, x^{(m)}\}$
- On a cross validation/test example x, predict

$(x^{(i)}_{test}, y^{(i)}_{test})$

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

$y = 0$

Possible evaluation metrics:

見Lec 11 p11, p14

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

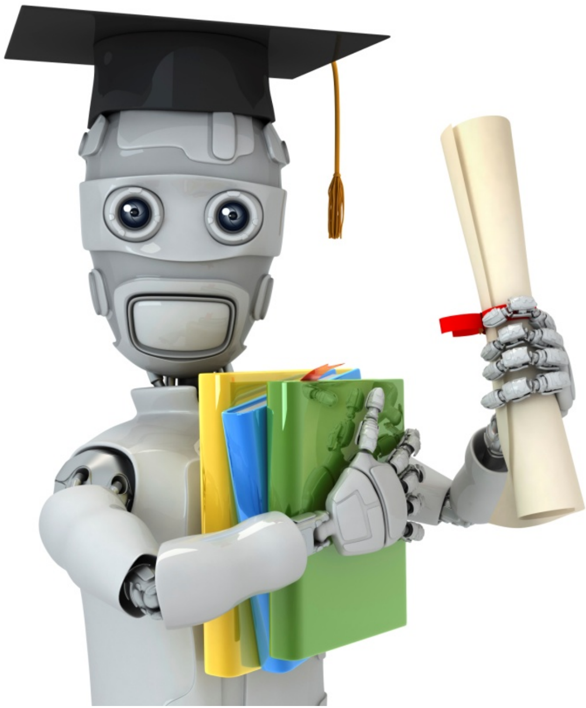
CV

Test set

一种选择参数ε的方法 就是你可以试一试 多个不同的ε的取值 然后选出一个 使得F1-积分的值最大的那个ε 也就是在交叉验证集中表现最好的

Can also use cross validation set to choose parameter ε

更一般来说当 我们需要作出决定时 比如要包括哪些特征 或者说要确定参数ε取多大合适 我们就可以 不断地用交叉验证集来评价这个算法 然后决定



Machine Learning

Anomaly detection

Anomaly detection vs. supervised learning

为什么我们不 直接用监督学习的方法呢？ 为什么不直接用 逻辑回归或者 神经网络的方法 来直接学习这些带标签的数据 从而给出预测 y=1 或 y=0 呢？

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. $\boxed{p(x)}$
- 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.

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.

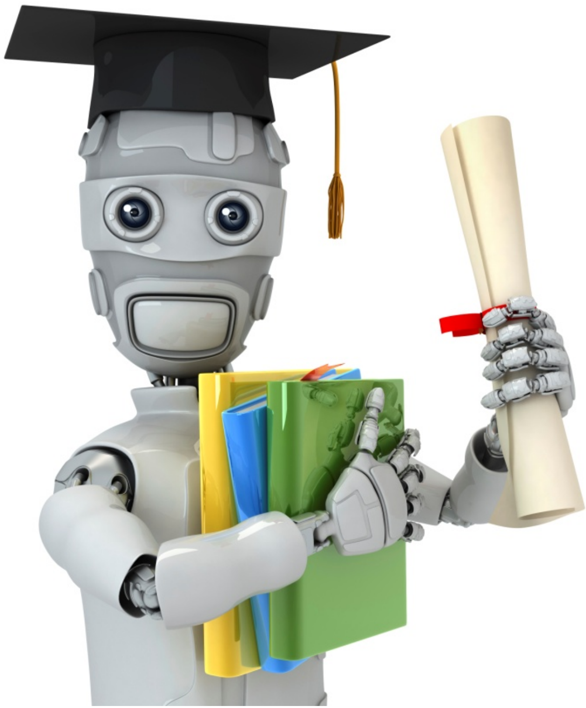
Spam

Anomaly detection vs. Supervised learning

Anomaly detection	Supervised learning
• Fraud detection $y=1$ 如果你掌握了大量的实施诈骗犯罪的人那么有时候欺诈检测的方法也可能会 偏向于使用监督学习算法	• Email spam classification 对于垃圾邮件的问题 我们通常有足够多的垃圾邮件的样本
• Manufacturing (e.g. aircraft engines)	• Weather prediction (sunny/rainy/etc).
• Monitoring machines in a data center	• Cancer classification
⋮	⋮

Anomaly detection

Choosing what features to use



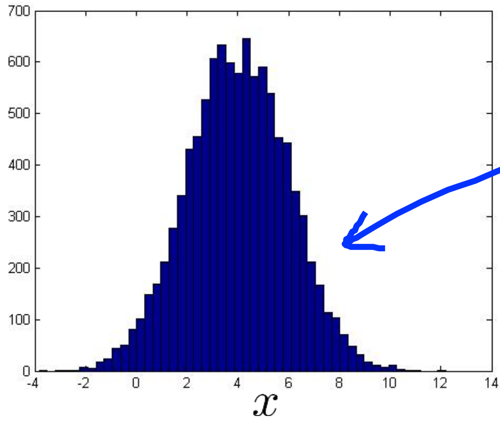
Machine Learning

用异常检测时 对它的效率 影响最大的 因素之一是 你使用什么features

Non-gaussian features

如果我有一个特征变量 比如 x1 直方图是这样的 那么我就用 x1 的对数 log(x1) 来替换掉 x1 所以 经过替换 这就是我的新 x1 我把它的直方图画在右边 这看起来更像高斯分布了

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



hist

除了取对数变换之外
还有别的一些方法
也可以用

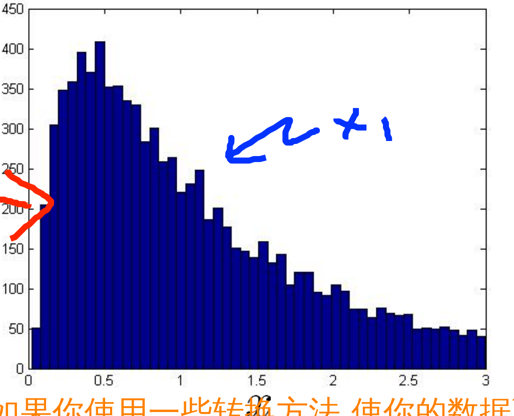
$x_1 \leftarrow \log(x_1)$

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

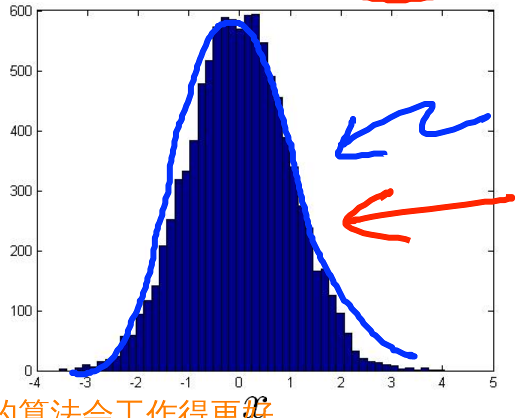
$x_3 \leftarrow \sqrt{x_3} = x_3^{\frac{1}{2}}$

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

如果我的数据是这样的话 通常我要做的事情 是对数据进行一些不同的转换 来确保这些数据 看起来更像高斯分布 虽然通常来说你不这么做 算法也会运行地很好 但如果你使用一些转换方法 使你的数据更像高斯分布的话 你的算法会工作得更好



$\log(x)$

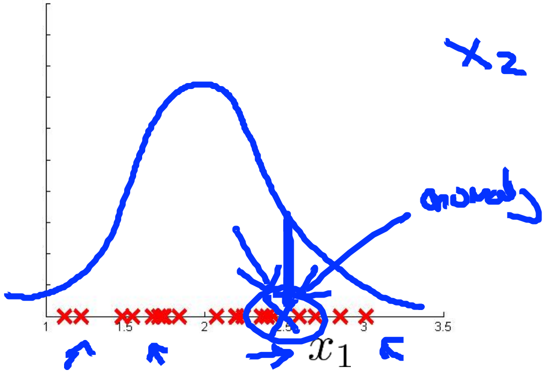


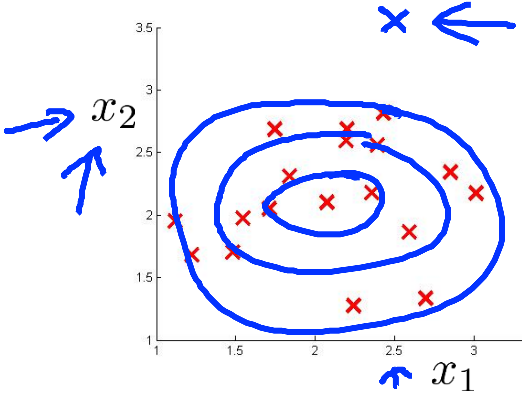
这跟我们之前 学习监督学习算法时的 误差分析步骤是类似的 也就是说 我们先完整地训练出 一个学习算法 然后在一组交叉验证集上运行算法 然后找出那些预测出错的样本 然后再看看 我们能否找到一些其他的特征变量 来帮助学习算法 让它在那些交叉验证时判断出错的样本中表现更

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





(看不清楚無所謂)
假如我的异常样本中 x 的取值为2.5 因此 我画出我的异常样本 你不难发现 它看起来就像被淹没在 一堆正常样本中似的

能不能启发我 想出一个新的特征 x2 来帮助算法区别出 不好的样本

Monitoring computers in a data center

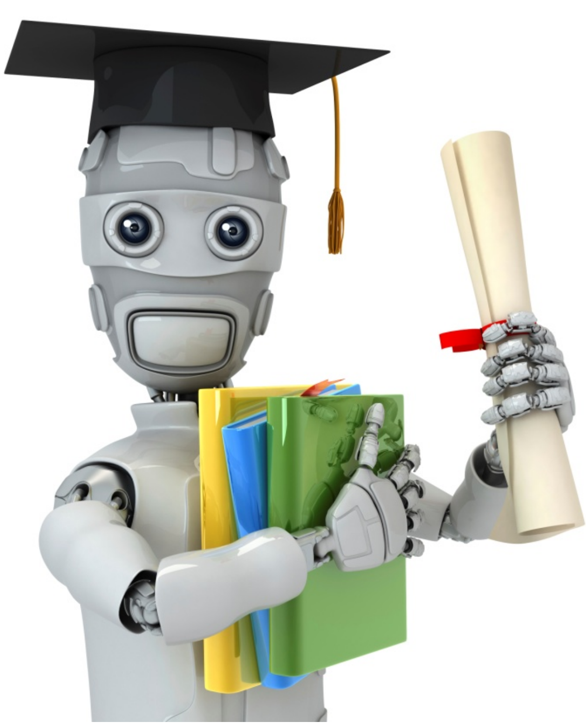
- Choose features that might take on unusually large or small values in the event of an anomaly.
  - x_1 = memory use of computer
  - x_2 = number of disk accesses/sec
  - x_3 = CPU load
  - x_4 = network traffic

x_5和x_6都可以

我怀疑其中一个出错的情形
是我的计算机在执行一个
任务时 进入了一个死循环
因此CPU负载升高
但网络流量没有升高

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

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



Machine Learning

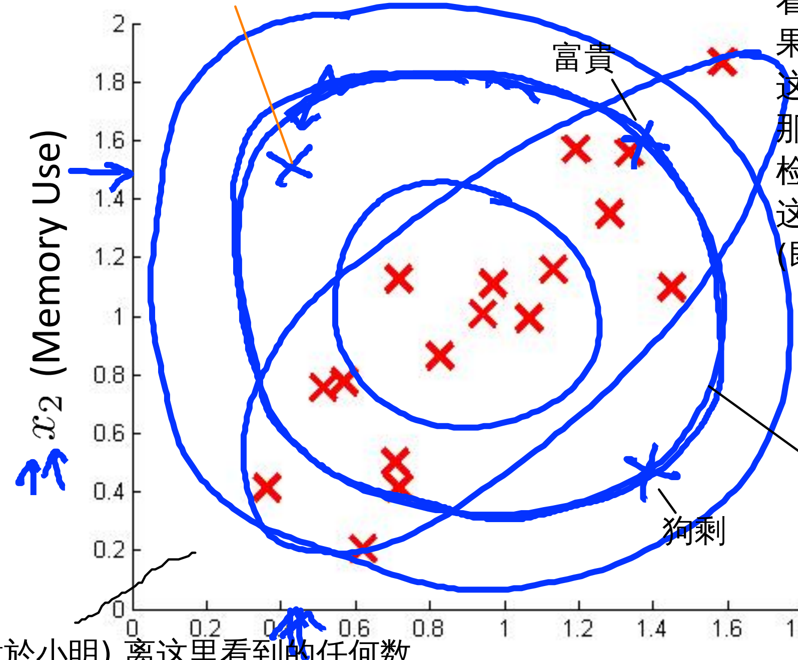
Anomaly detection

Multivariate Gaussian distribution

multivariate Gaussian distribution它有一些优势 也有一些劣势 它能捕捉到一些之前的算法检测不出来的异常

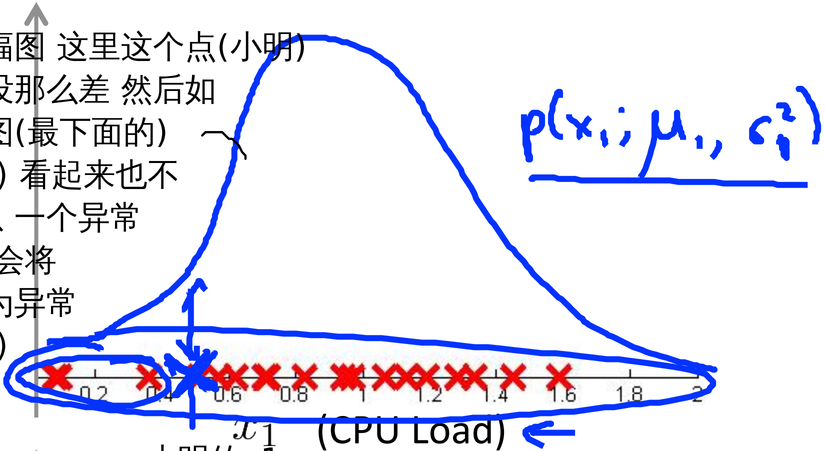
Motivating example: Monitoring machines in a data center

如果我們看這點(0.4, 1.5)
(叫他小明)

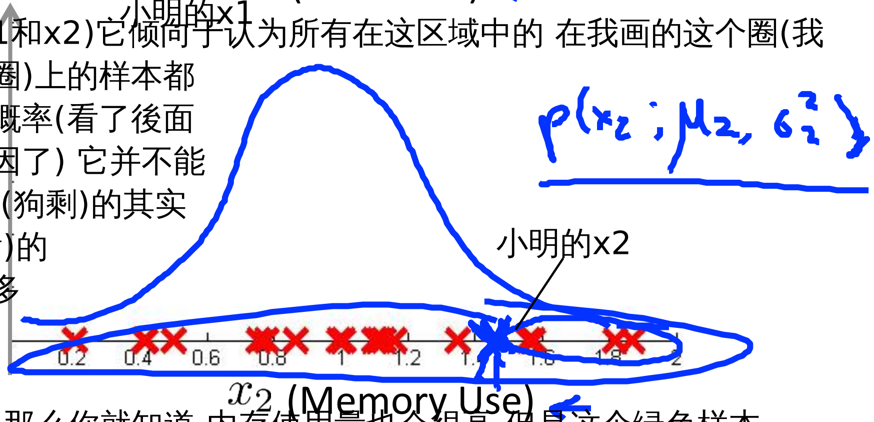


x_1 (CPU Load)

如果你看这幅图 这里这个点(小明) 看起来它并没那么差 然后如果你看这幅图(最下面的) 这个叉(小明) 看起来也不那么差. 所以 一个异常检测算法 不会将这个点标记为异常 (即判斷錯了)



(若分别看x1和x2)它倾向于认为所有在这区域中的 在我画的这个圈(我黑線所指的圈)上的样本都具有相同的概率(看了後面的就知道原因了) 它并不能意识到 这边(狗剩)的其实比那边(富貴)的概率要低得多



(對於小明) 离这里看到的任何数据都很远 看起来它应该被当做一个异常数据 所以我的 好的样本的数据 看起来 CPU 负载 和内存使用量 是彼此线性增长的关系 所以 如果我有一台机器 CPU 使用量很高 那么你就知道 内存使用量也会很高 但是这个绿色样本看起来 CPU 负载很低 但是内存使用量很高 我以前从没在 训练集中见过这样的 看起来它应该是异常的

Andrew Ng

所以 为了解决这个问题 我们要开发一种 改良版的异常检测算法 叫做多元高斯分布或者多元正态分布的东西

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)

x, μ都是n維向量, Σ是n×n矩陣

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

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

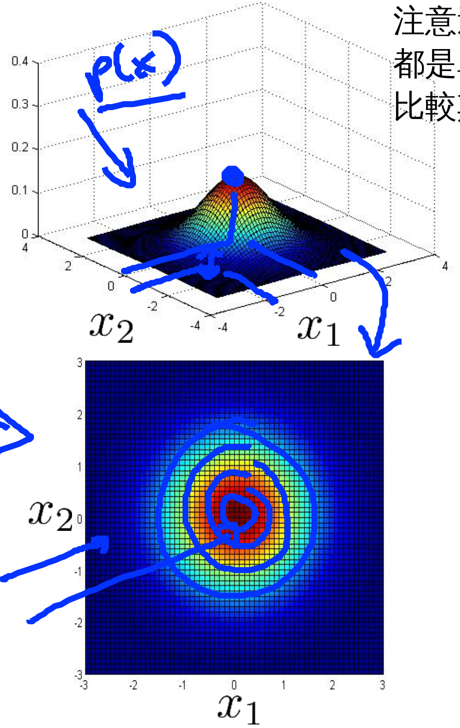
Multivariate Gaussian (Normal) examples

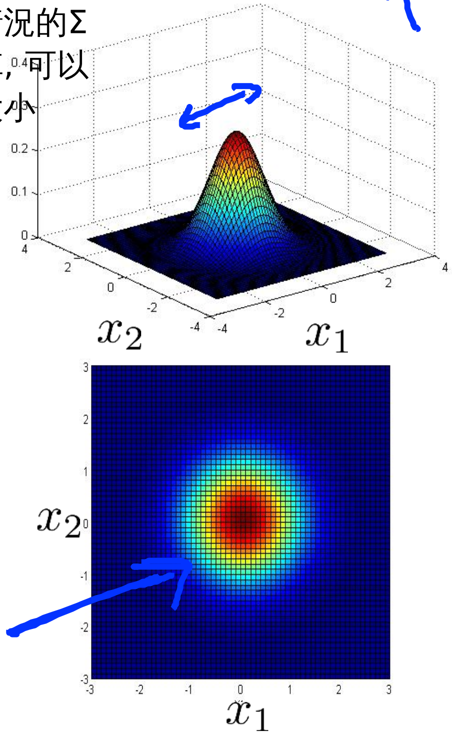
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \boxed{\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}$$

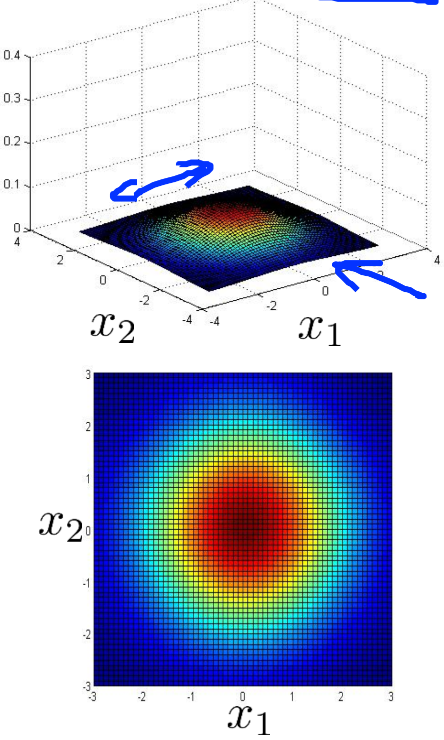
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \boxed{\Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix}}$$

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

注意這三種情況的Σ
都是單位矩陣，可以
比較其元素大小



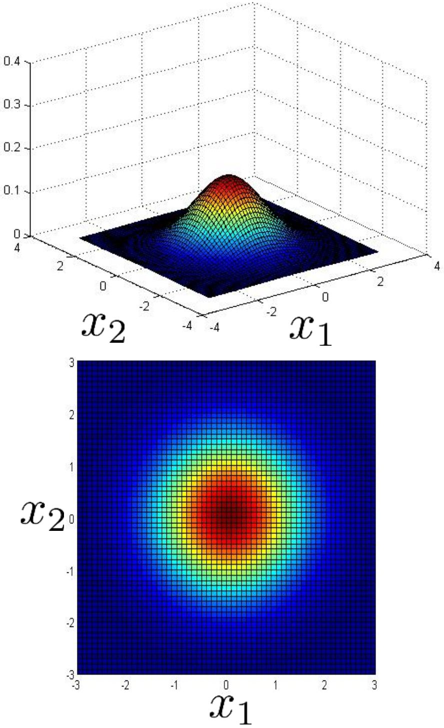




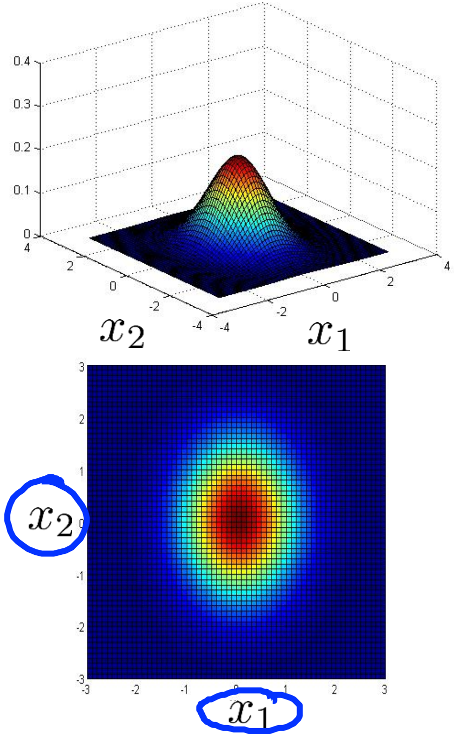
Andrew Ng

Multivariate Gaussian (Normal) examples

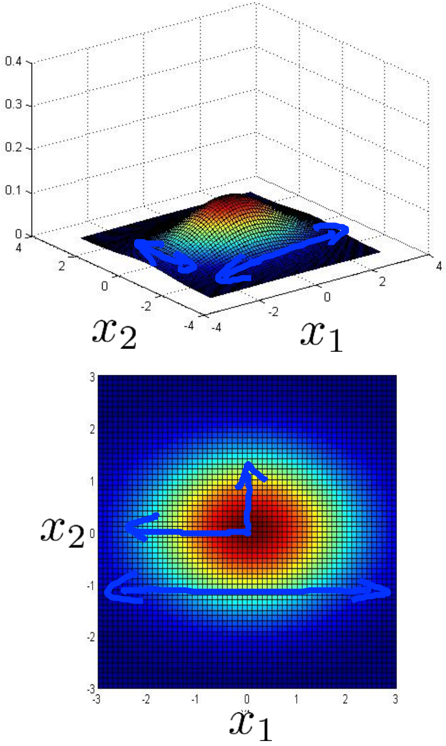
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \ \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



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



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

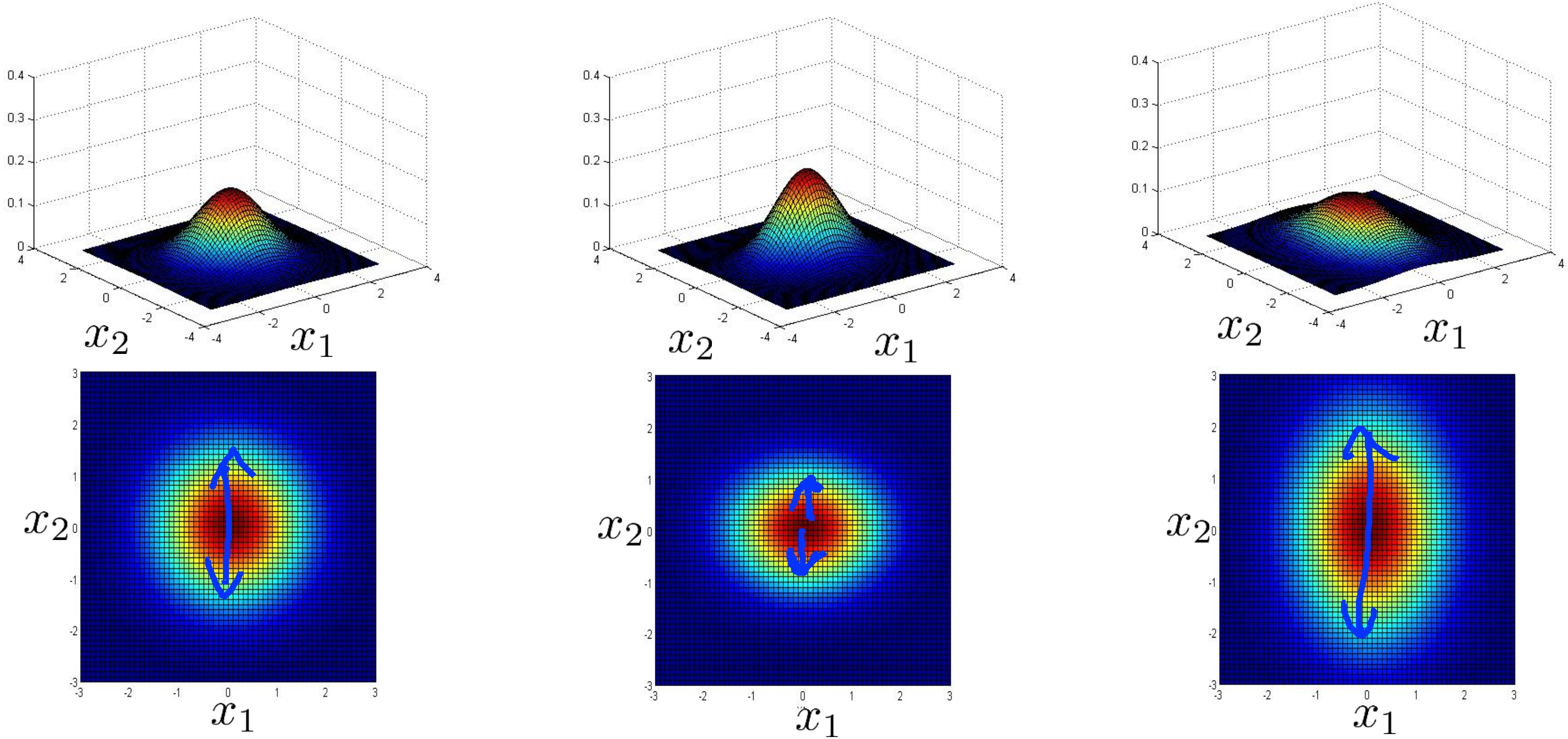


Multivariate Gaussian (Normal) examples

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

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

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

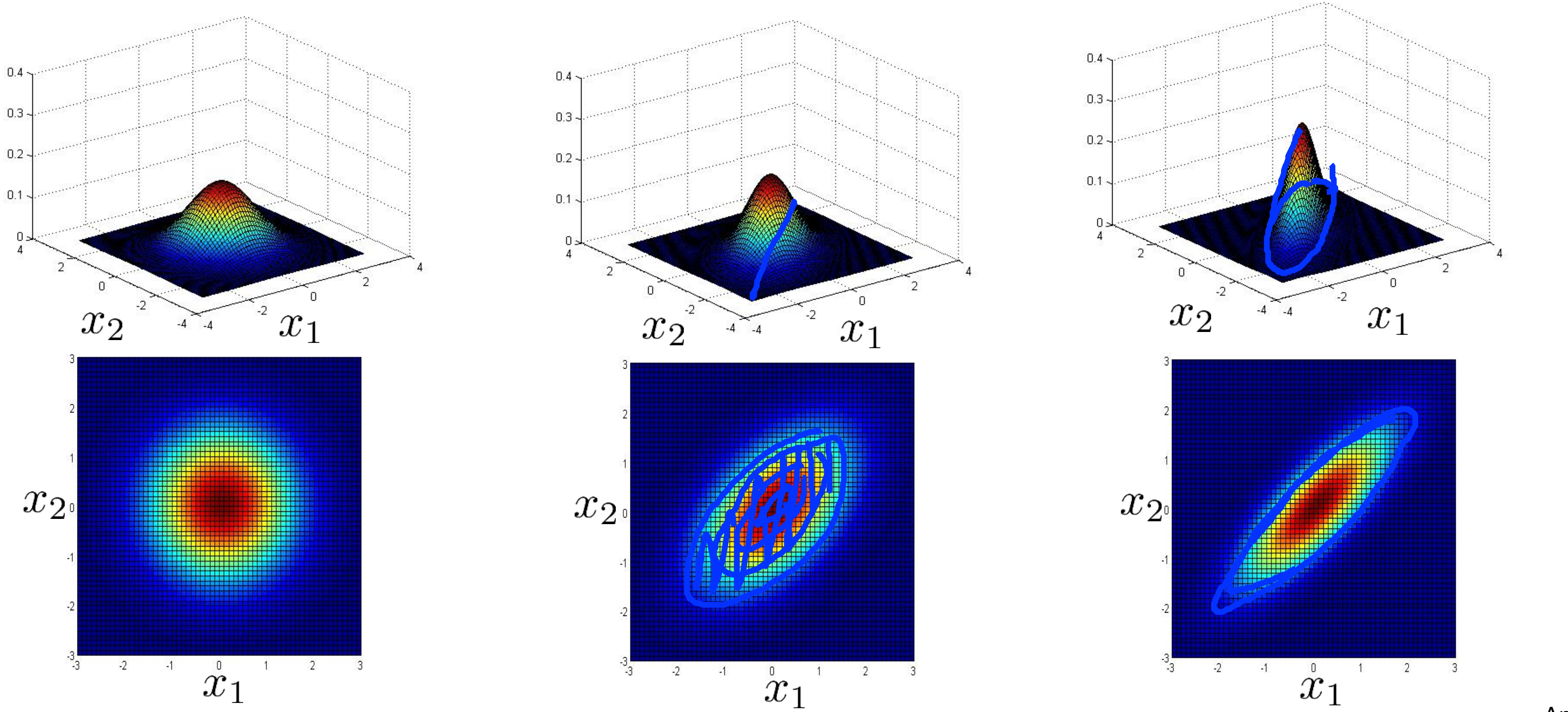


Multivariate Gaussian (Normal) examples

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

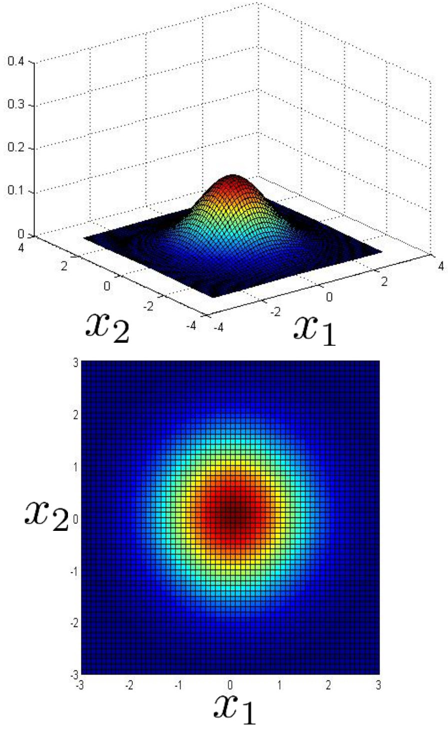
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

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

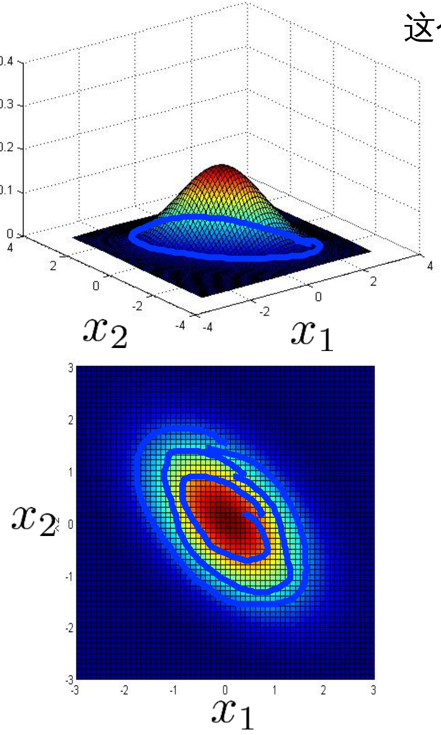


Multivariate Gaussian (Normal) examples

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

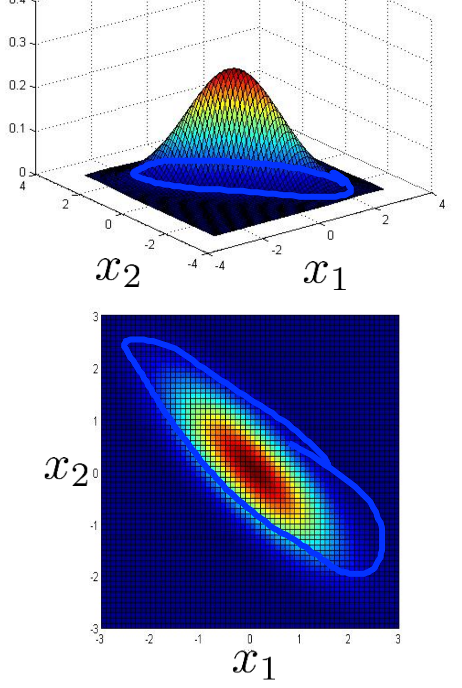


$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$$



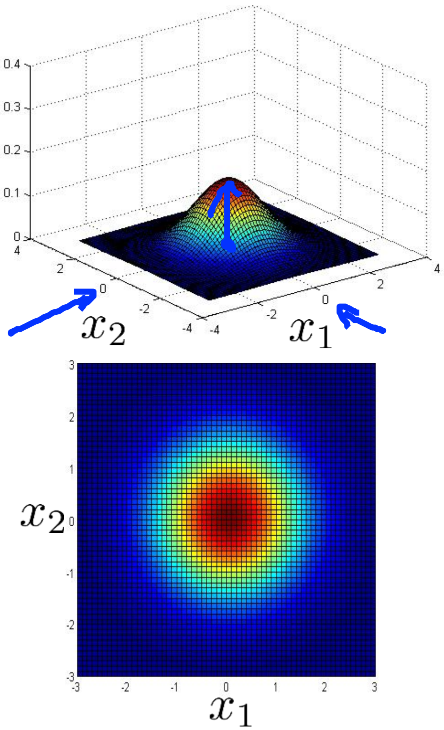
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}$$

这个捕捉到了 x1 和 x2 的负相关性

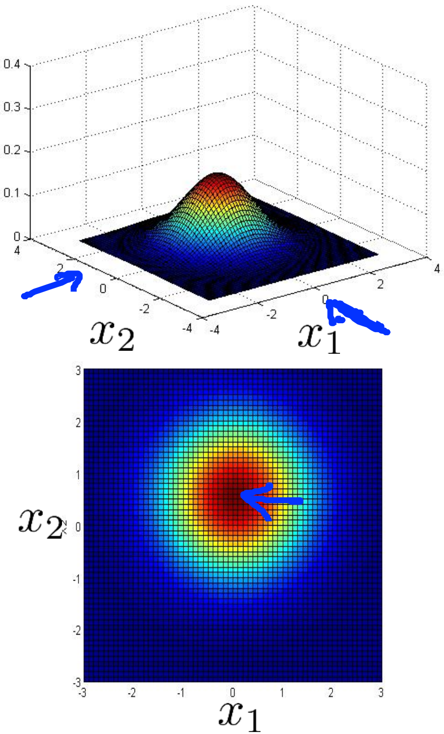


Multivariate Gaussian (Normal) examples

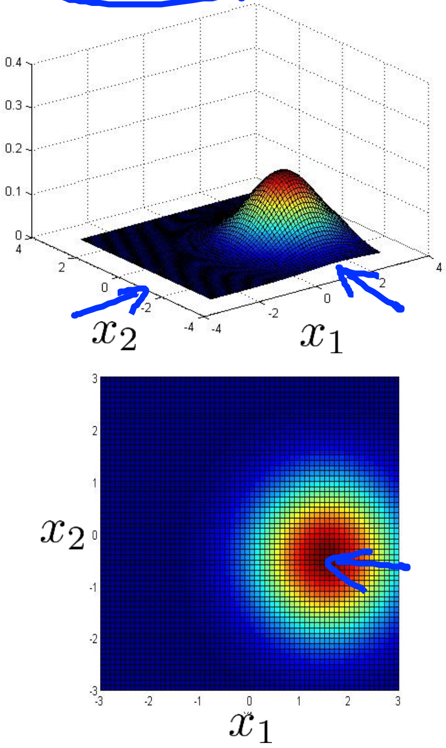
$$\boxed{\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}} \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

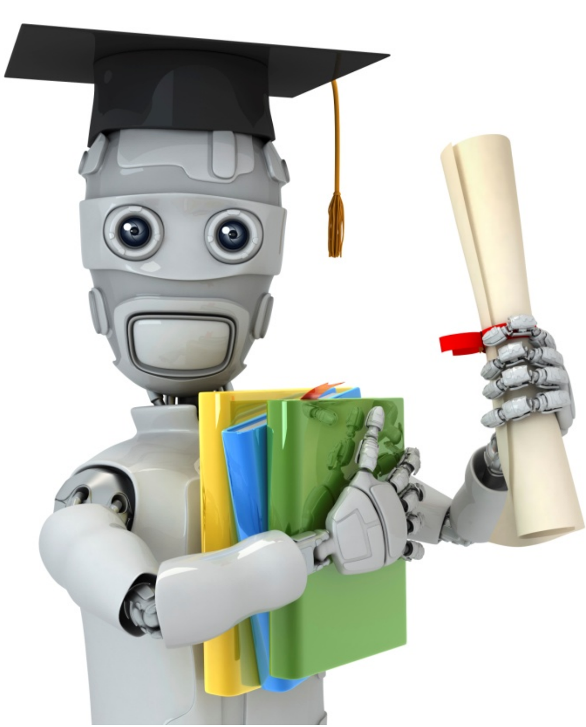


$$\boxed{\mu = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}} \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



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





Machine Learning

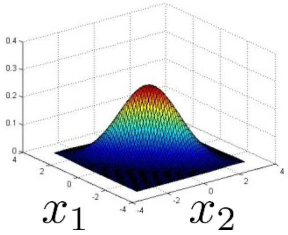
Anomaly detection

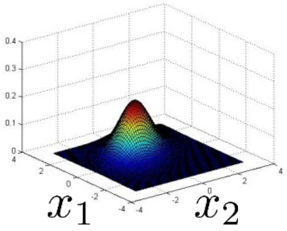
Anomaly detection using the multivariate Gaussian distribution

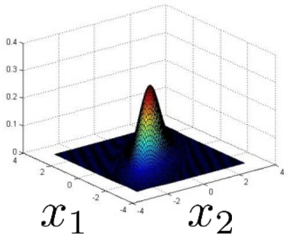
Multivariate Gaussian (Normal) distribution

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

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





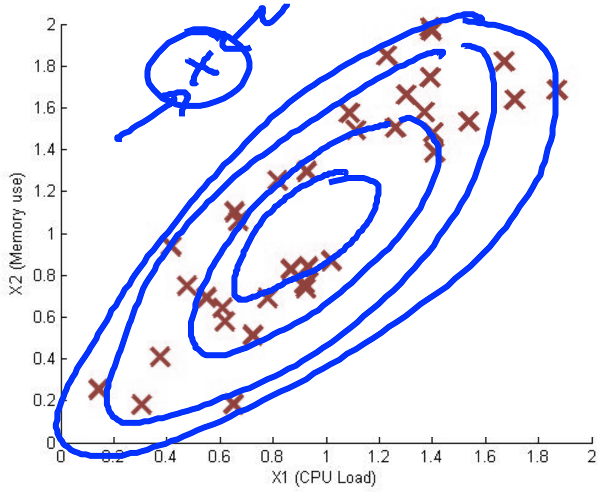


Parameter fitting:

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

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

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

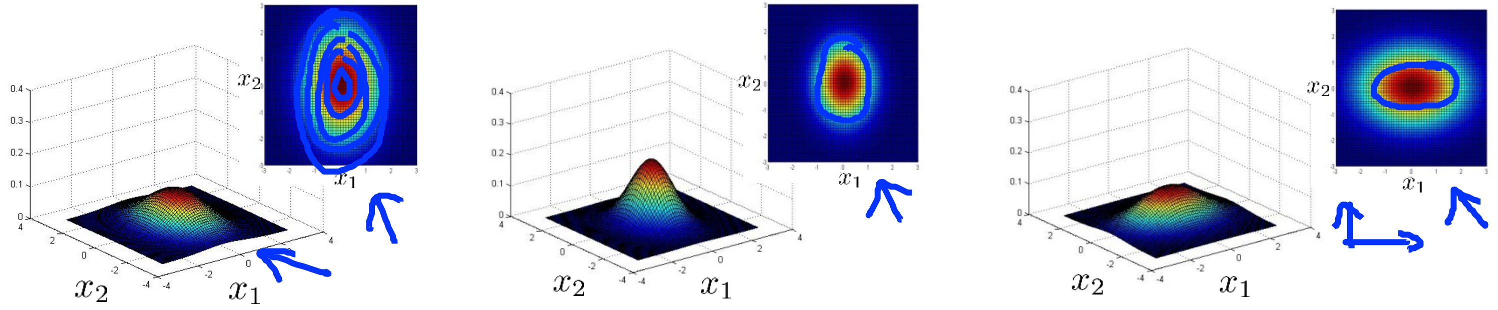
$$p(x) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

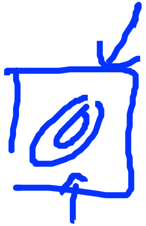
Flag an anomaly if $p(x) < \varepsilon$

Relationship to original model

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

$p(x)$





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

Corresponds to multivariate Gaussian

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

where $\Sigma =$

在实际应用当中 左边这个原来的模型比较常用

典型的经验法则是 我**只**在当 m 远大于 n 的时候使用多元高斯模型

Original model vs. Multivariate Gaussian

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

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

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

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

Automatically captures correlations between features

Tao:注意前面的例子中, CPU 负载和内存使用量是正相關的

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

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

Computationally more expensive

Σ $\sim \frac{n^2}{2}$

$x_1 = \cancel{x_2}$

$\cancel{x_3} = x_4 + x_5$

OK even if m (training set size) is small

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

$\boxed{m \geq 10n}$

Andrew Ng