

异常检测 (Anomaly Detection)

异常检测做法：

1. 对既存数据进行建模，也就是对 x 的分布概率进行建模，得到模型 $p(x)$ 。
2. 设置一个阈值 ϵ ，当新的数据 x 在 $p(x)$ 模型中的概率低于阈值时，我们就将之标记为异常。

异常检测的栗子：

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 $\underline{p(x) < \epsilon}$

→ 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.

...

$p(x) < \epsilon$

x_1
 x_2
 x_3
 x_4
 $p(x)$

欺诈检测：对用户行为特征进行建模，当用户行为异常时判断用户存在欺诈行为或者被盗号

具体来说， x_1 也许是用户登陆的频率， x_2 也许是用户访问某个页面的次数或者交易次数， x_3 也许是用户在论坛上发贴的次数， x_4 是用户的打字速度（有些网站是可以记录用户每秒打了多少个字母的），因此你可以根据这些数据建一个模型 $p(x)$ 。

最后你将得到模型 $p(x)$ ，然后你可以用它来发现你网站上的行为奇怪的用户，你只需要看哪些用户的 $p(x)$ 概率小于 ϵ ，然后拿这些用户的档案做进一步筛选。

工业检测：例如生产飞机引擎，对之前的产品进行建模，当新引擎的数据异常，则需要做额外的测试

高斯分布 (Gaussian Distribution) 也叫正态分布 (Normal Distribution)

高斯分布的概率密度公式是见下图，其实我们并不需要记住这个公式，它只是左边这条钟形曲线对应的公式。我们没有必要记住它，当我们真的需要用到它时我们总可以查资料找到它。

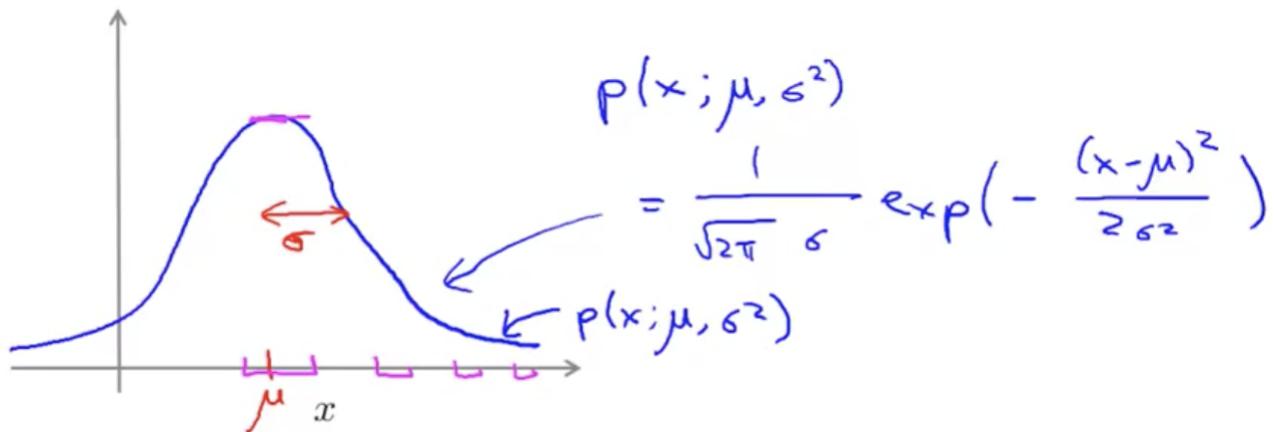
μ (读作miu) 决定高斯分布的中心，标准差 σ (读作sigma) 决定高斯分布的宽度

Gaussian (Normal) distribution

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

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

π "distributed as"



下图中，给定一组数据（x轴），我们可以估算出钟形曲线，这一步称为参数估计，也就是估计 μ 和 σ 的值

图的下方就是参数估计的标准计算公式，

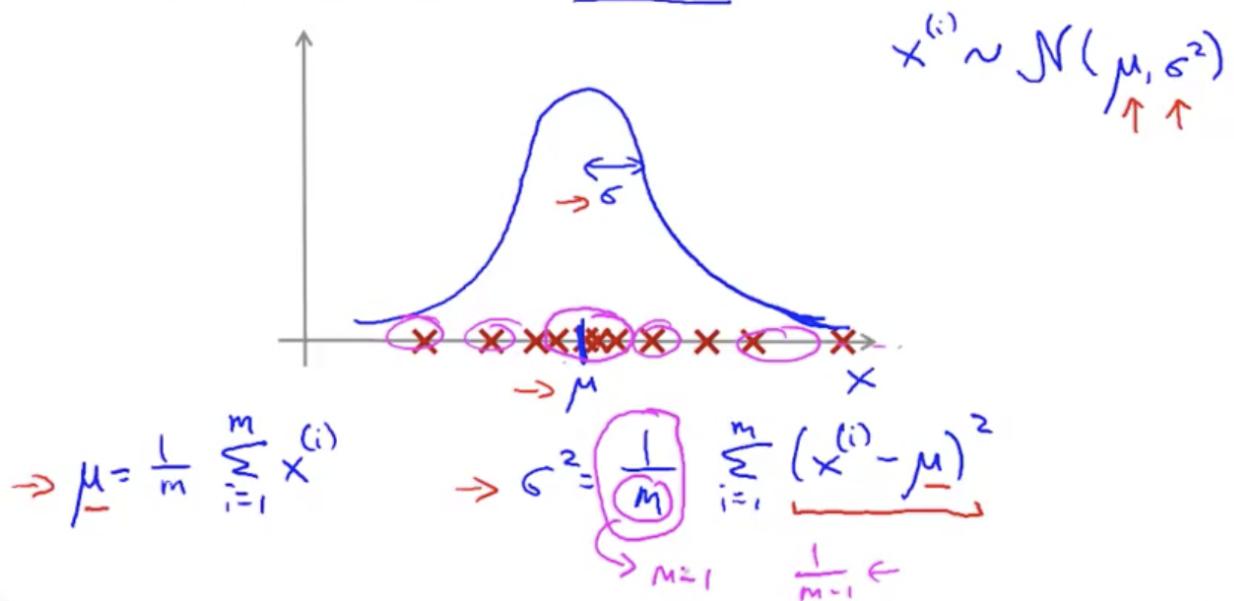
我们估计 μ 的方法是，对所有样本求平均值， μ 就是平均值参数，因此对训练集中的 m 个样本取平均值就得到了高斯分布的中心位置

σ 平方表示方差，先将所有的样本从 $x(1)$ 到 $x(m)$ 减去平均值 μ ，平方再求和，实际上 μ 是用之前的这个公式计算出来的。而方差的含义或者说至少一种方差的定义，是将这一项所有样本的差值平方和，再求平均

用统计学术语来说，这里其实就是对 μ 和 σ 平方的极大似然估计 (maximum likelihood) 虽然在理论上求 σ 公式中的第一项使用 m 还是 $m-1$ 分之一是一个值得探讨的问题 (理论特性和数学性质上稍有不同)，但在实践中这两者并没有什么太大的区别

Parameter estimation

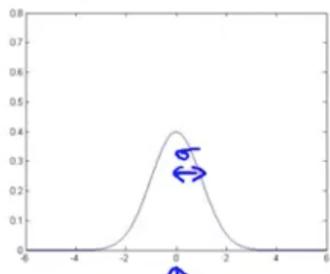
→ Dataset: $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ $x^{(i)} \in \mathbb{R}$



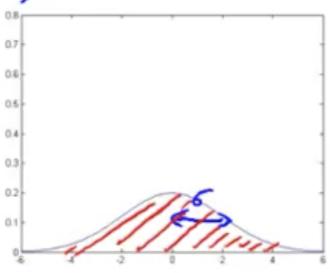
关于高斯分布的一些栗子,

Gaussian distribution example

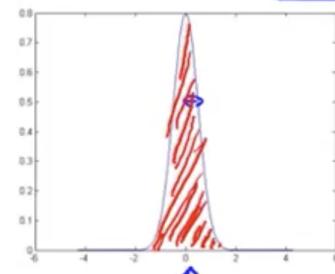
$$\rightarrow \mu = 0, \sigma = 1$$



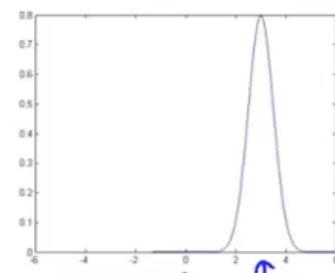
$$\rightarrow \mu = 0, \sigma = 2$$



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



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



$$\sigma^2 = 0.25$$

Andrew N

与SVM的特征曲线不同, 这里的顶点并不是固定的, 但是参数 μ 和 σ 的变动特征还是一样的,
 μ 决定中心点, σ 决定曲线的高矮胖瘦

Algorithm

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 $\mu_1, \dots, \mu_n, \sigma_1^2, \dots, \sigma_n^2$
 - $\mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)}$ $p(x_j; \mu_j, \sigma_j^2)$ $\mu_1, \mu_2, \dots, \mu_n$
 $\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix} = \frac{1}{m} \sum_{i=1}^m x^{(i)}$
 - $\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)} - \mu_j)^2$
- 3. Given new example x , compute $p(x)$:
 - $p(x) = \prod_{j=1}^n p(x_j; \mu_j, \sigma_j^2) = \prod_{j=1}^n \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}\right)$

上图总结了异常检测算法的各个步骤，

第一步，选择特征。

找出一些我们认为的能够比较出反常样本的特征 x_i ，通常的做法是尽可能选择能够描述数据相关属性的特征。

第二步，给出一组 m 个无标签数据构成的训练集，从 $x(1)$ 到 $x(m)$ 我们要拟合出 μ_1 到 μ_n 以及方差值 $(\sigma_1)^2$ 到 $(\sigma_n)^2$ 的值。

更具体来说， μ_j 是特征 j 的平均值，因此 μ_j 对应的模型就是 $p(x_j; \mu_j, (\sigma_j)^2)$ ，因此 μ_j 就相当于对特征 j 的所有训练集数据取平均值。同时，我们需要对 j 从 1 到 n 计算这些概率值，也就是说用这些公式来估计 μ_1 再估计 μ_2 直到 μ_n 。同样地，对于 σ^2 也一样，同时，用向量化的方法也可以写出来。所以，可以把 μ 假想成一个向量 那么向量 μ 就有 $\mu_1 \mu_2$ 直到 μ_n ，那么这个公式的向量表示形式就能被写出来， μ 的值等于 $x(i)$ 的值从 $i = 1$ 到 n 求和，再乘以 $1/m$ ，其中 $x(i)$ 是一系列特征组成的向量，同时包含了所有 n 个值

同样地，我们也可以写出估计 (σ_j^2) 的向量化的公式。

最后，大写字母 Pi ，表示一系列数值的乘积，给出一个新的样本，我们要做的就是计算出 $p(x)$ 的值来。

Building an Anomaly Detection System

下图中，想要评估一个异常检测算法，首先我们要用训练集拟合出模型 $p(x)$

然后用交叉验证集和测试集来预测 y 的数值，这里由于标签会产生偏斜，因为 $y = 0$ 远远多于 $y = 1$ 的情况，

因此我们会用到在第6周提到的诊断学习算法所用到的 F1 积分

最后，阈值 ϵ 用来决定什么时候把一个样本判定为异常，如果我们有一组交叉验证集，那么选择参数 ϵ 最好的办法就是尝试多个不同的 ϵ 取值，然后选出使得 F1 积分的值最大的那个，也就是在交叉验证集中表现最好的 ϵ

Algorithm evaluation

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

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

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

那么既然用到了y最为标签，为什么不干脆用监督学习算法呢

Anomaly detection vs. Supervised learning

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

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 ↵

上图中我们可以看到，使用哪个算法主要取决于以下几点：

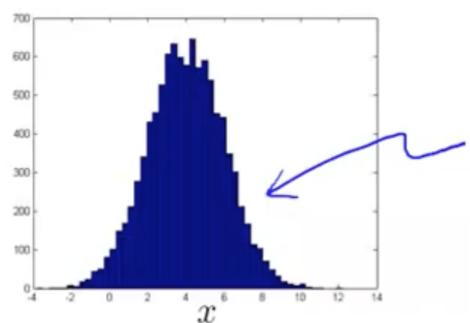
第一，**样本数量**。如果有20-50组正样本，即我们的正样本数量很小，并且我们的负样本（即正常的数据）数量很大，那么我们应该考虑使用异常检测算法

反过来说，如果正负样本数量都比较大，那么应该考虑使用监督学习算法

第二，**异常种类**。当我们无法确定异常的种类，例如出现故障的原因有非常多的并且无法确定的情况下，应该是用异常检测算法，反过来说，如果我们对正样本的出现形式很熟悉，并且能预测到未来会出现的正样本与训练集中的非常相似，那么监督学习算法将更加合理。

最后，也是最重要的一点，如何选择特征？

Non-gaussian features



$$p(x_i; \mu_i, \sigma^2_i)$$

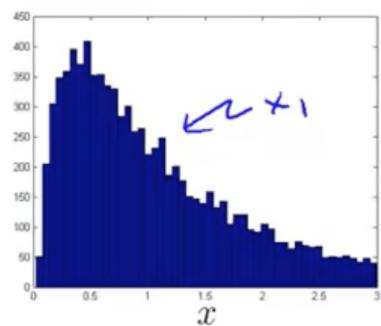
hist

$$x_1 \leftarrow \log(x_i)$$

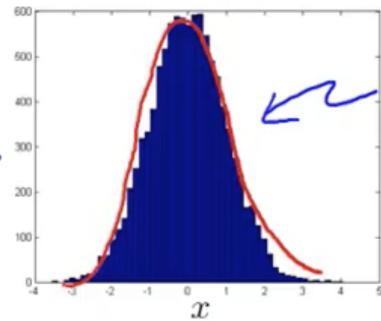
$$x_2 \leftarrow \log(x_1 + 1)$$

$$x_3 \leftarrow \sqrt{x_2} = x_2^{\frac{1}{2}}$$

$$x_4 \leftarrow \sqrt[3]{x_3} =$$

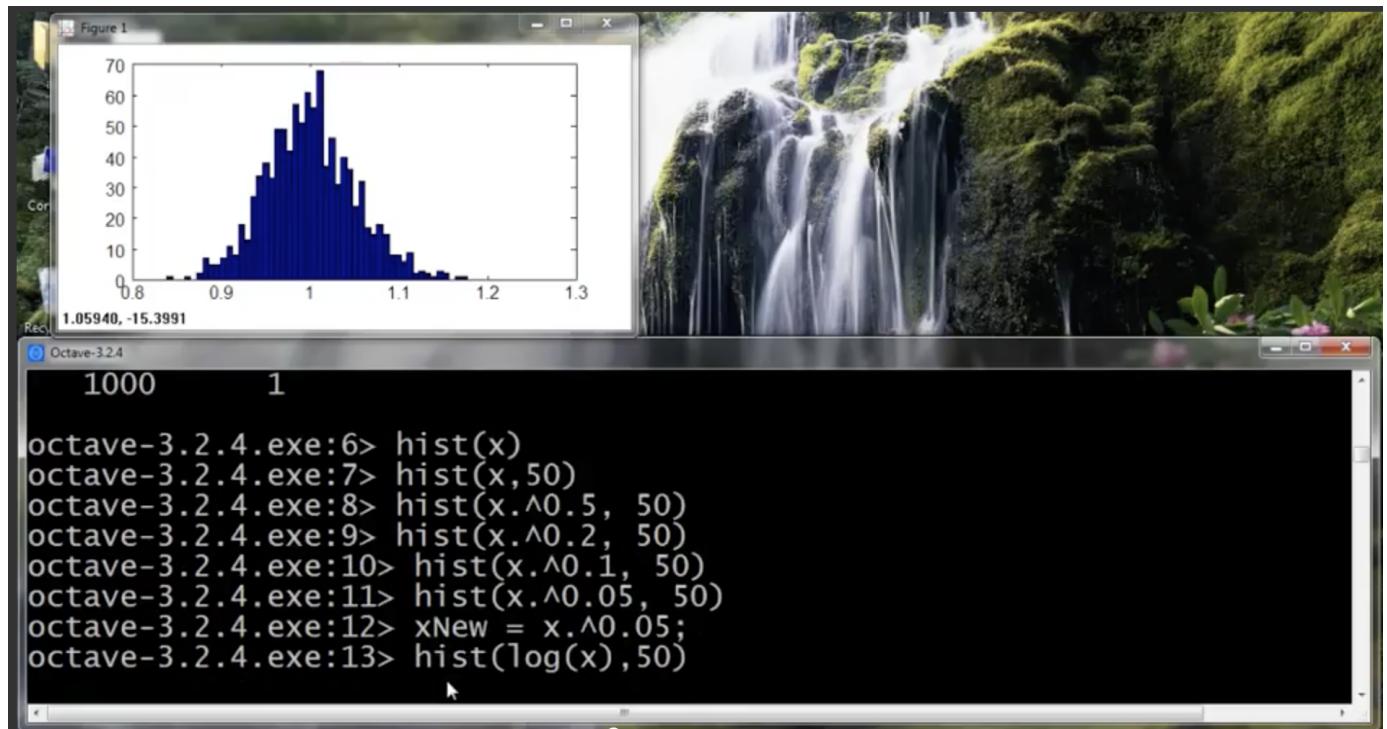


$$\log(x)$$



上图中左上角的图是一个理想的高斯分布，但是如果我们的特征看上去更像是左下角的图时该怎么办呢？

这种情况下我们可以想办法对特征进行转换，以使它们形成一个接近高斯分布的形状，例如上图中对 $x_1 \sim x_4$ 的处理



这里我们可以在Octave中使用`hist(x,50)`来将特征 x 表示为50个柱状组成的图形

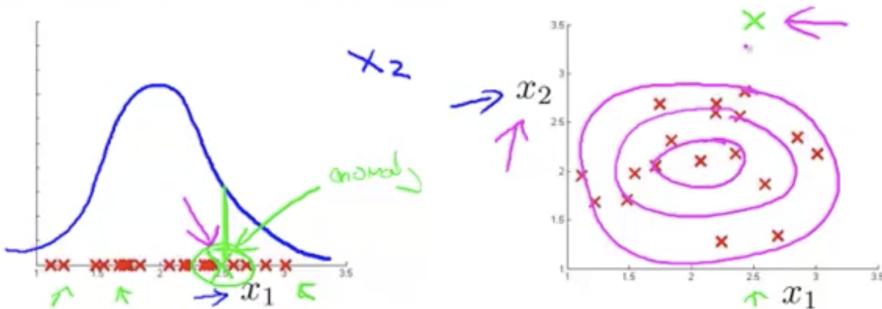
然后我们可以通过对 x 进行一系列的处理比如开根号，取对数等等，并最终得到一个接近高斯分布的特征集 x_{New}

→ 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



还有一个问题，那就是当异常样本非常接近阈值，以至于他混在一堆正常的样本中无法被发现，这里我们就希望能通过新的特征来将其排除在外，主要方法是通过分析异常样本，来得出一个新的特征，

比如下图中关于CPU监控中心的栗子

网络数据大量存在时，会导致CPU负荷上升，因此光看CPU负荷的数据无法判断是否异常，但是如果网络数据并不多，而CPU负荷仍然很高，则说明CPU确实异常了，在这里我们就可以通过原有的四个特征，得出新的特征x5，甚至x6，就是用CPU负荷或者CPU负荷的平方去除以网络数据流量，这样就能明显判断出那些数据异常了。

→ 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 ←

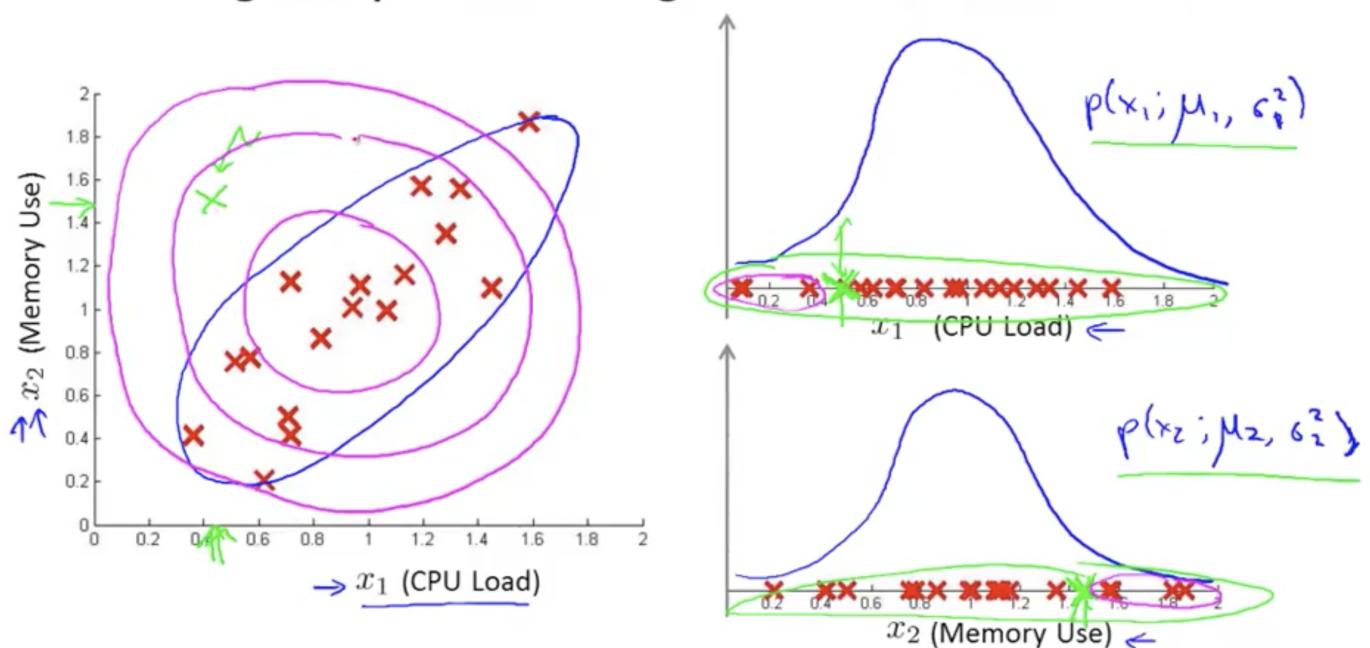
$$x_5 = \frac{\text{CPU load}}{\text{network traffic}}$$

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

【拓展内容】多元高斯分布

普通的高斯分布将无法对应下图中绿色x这样的情况，因为算法会倾向于认为粉色圈中的数据都是差不多的概率，从而忽略了显然是异常的绿色x样本

Motivating example: Monitoring machines in a data center



为了解决这个问题，我们需要对异常检测算法进行改良，这里就要用到多元高斯分布，见下图

Multivariate Gaussian (Normal) distribution

$\rightarrow x \in \mathbb{R}^n$. 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)

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

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

这里我们不再分别对各个特征建模，而是将其看作一个整体，就是一次性建立 $p(x)$ 模型

多元高斯分布的参数 包括向量 μ 和一个 $n \times n$ 的矩阵 Σ (即协方差矩阵，不是求和符号)

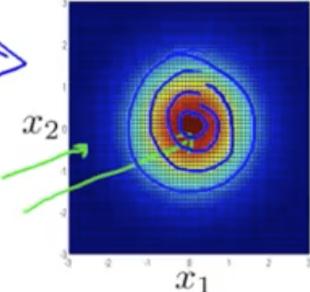
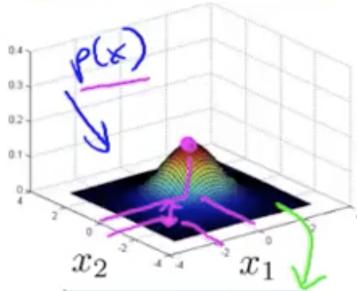
Σ 的绝对值叫做 Σ 的行列式 (determinant)，它是一个矩阵的数学函数，在Octave里可以使用命令 `det(Sigma)` 来计算它

以下是一些多元高斯分布的二维栗子，有助于对多元高斯分布有个粗略的概念

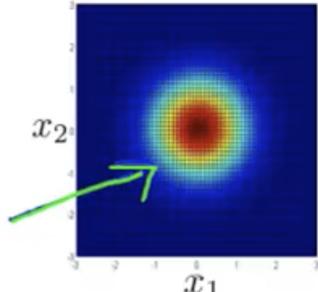
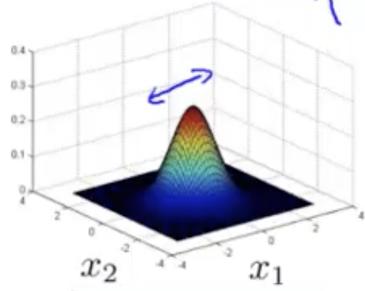
需要注意的是，由于钟形曲线体积的积分等于1，因此缩小 Σ 的平方就会得到一个瘦高的曲线，而增大 Σ 的平方就会得到一个矮胖的曲线

Multivariate Gaussian (Normal) examples

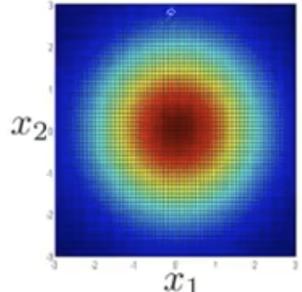
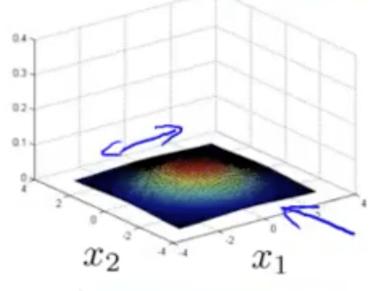
$$\rightarrow \mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



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

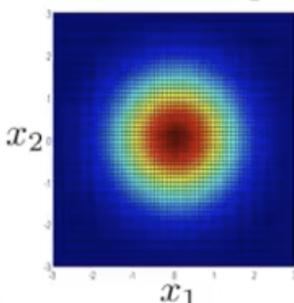
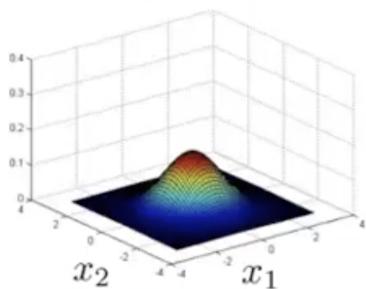


$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

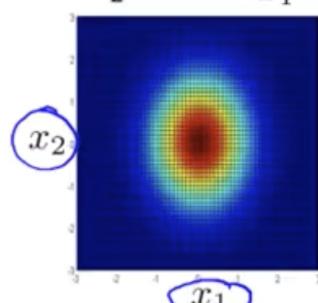
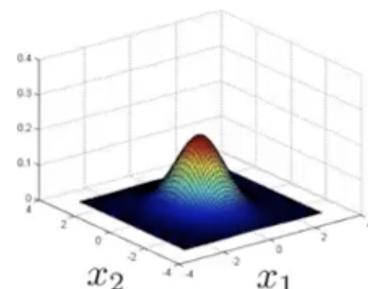


Multivariate Gaussian (Normal) examples

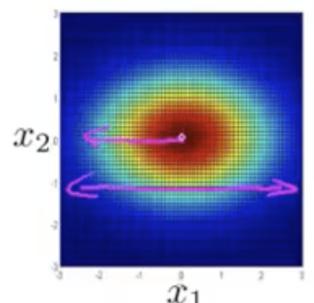
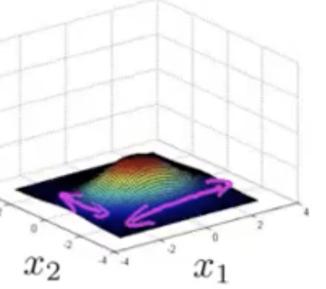
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



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



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

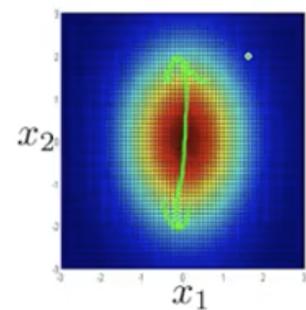
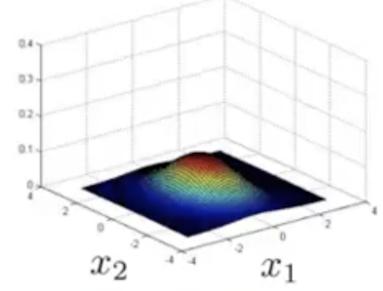
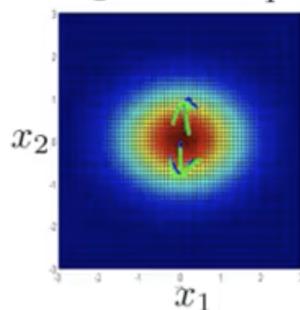
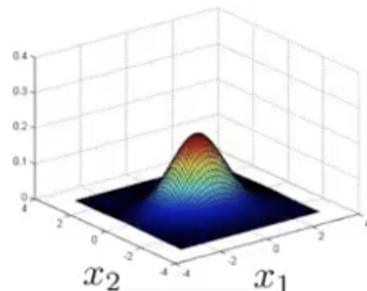
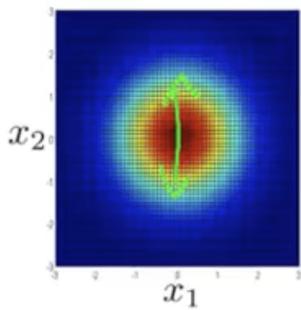
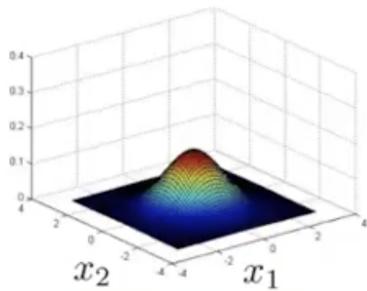


Multivariate Gaussian (Normal) examples

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

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

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



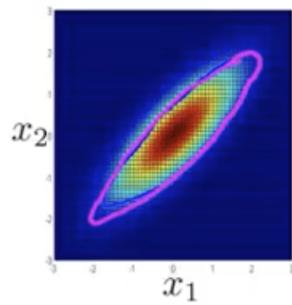
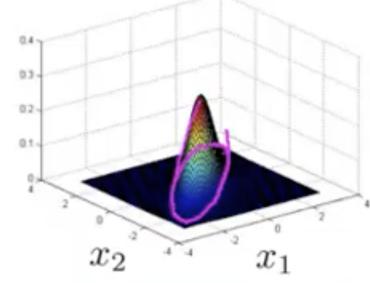
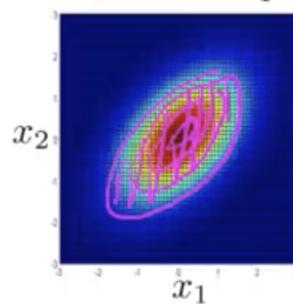
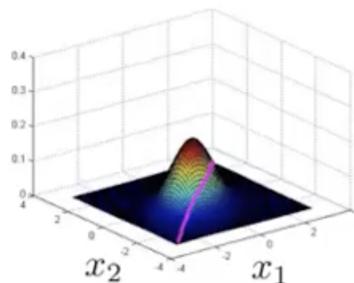
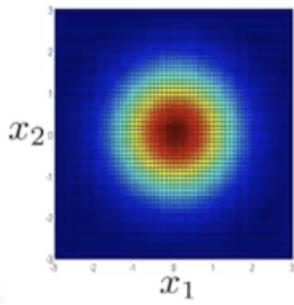
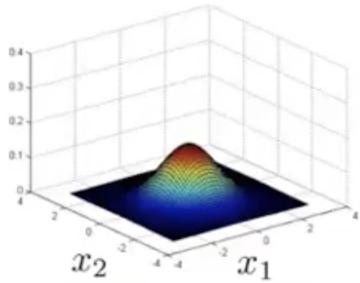
下两幅图中的栗子正是多元高斯分布的过人之处，通过对矩阵 Σ 的对角线进行定义，我们可以作出一个在二维上看来倾斜的曲线，这就完全避免了开头绿色x那样的情况

Multivariate Gaussian (Normal) examples

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

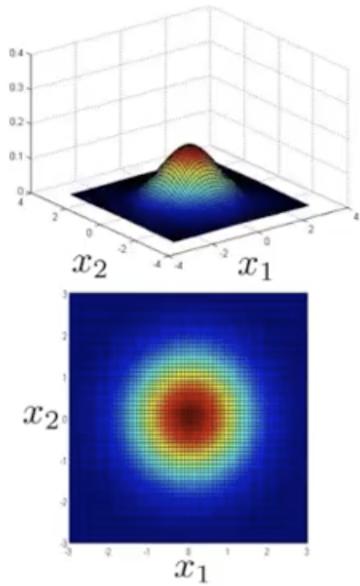
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

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

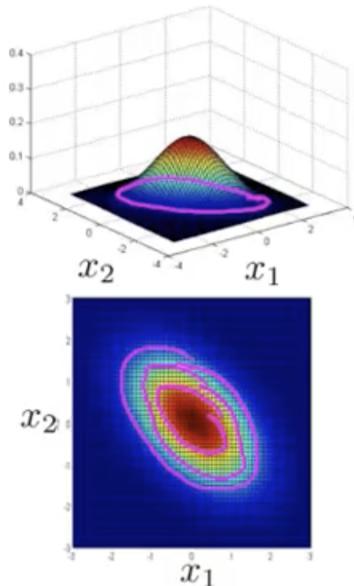


Multivariate Gaussian (Normal) examples

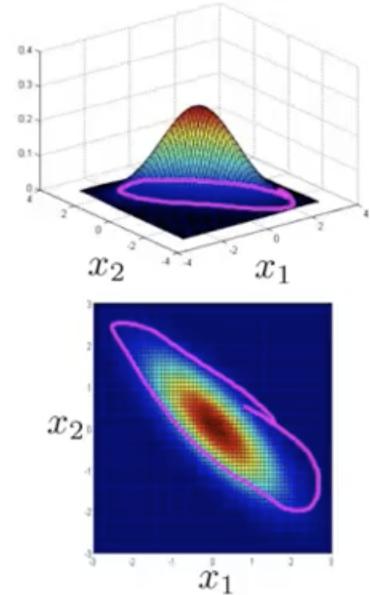
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



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

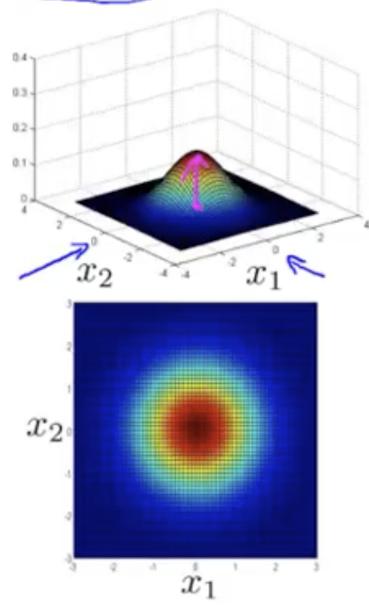


$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}$$

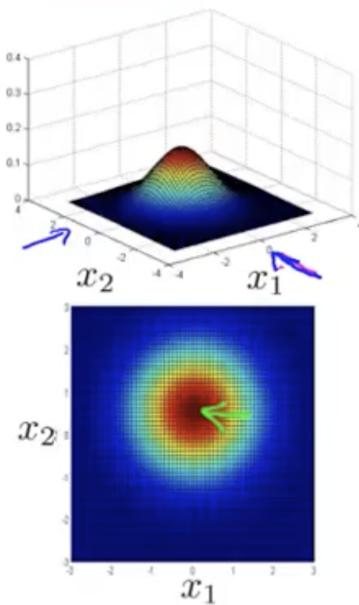


Multivariate Gaussian (Normal) examples

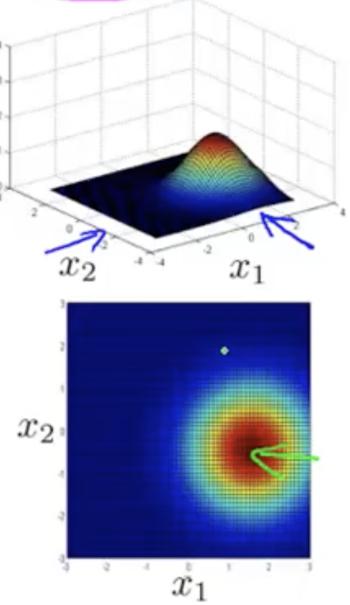
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



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



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



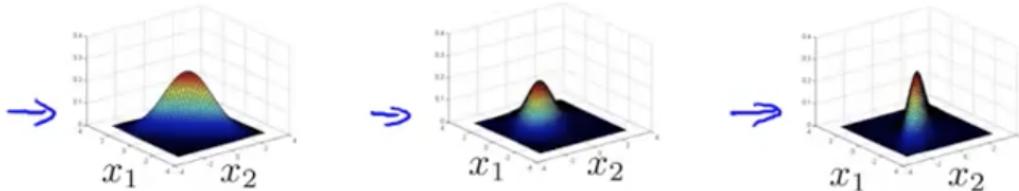
多元高斯分布的应用

Multivariate Gaussian (Normal) distribution

Parameters μ, Σ

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

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



Parameter fitting:

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

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

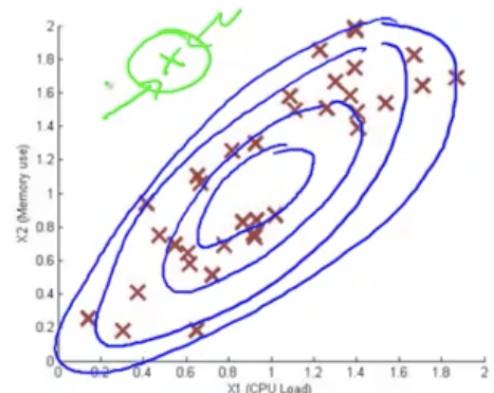
上图中罗列了估计多元高斯分布的参数 μ 和 Σ 的公式， μ 等于训练样本的平均值，而 Σ 的式子实际上就是PCA中那个式子

有了这些参数，我们就可以开始构造异常检测算法了

Anomaly detection with the multivariate Gaussian

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

$$\boxed{\begin{aligned} \mu &= \frac{1}{m} \sum_{i=1}^m x^{(i)} \\ \Sigma &= \frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu)(x^{(i)} - \mu)^T \end{aligned}}$$



2. Given a new example x , compute

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

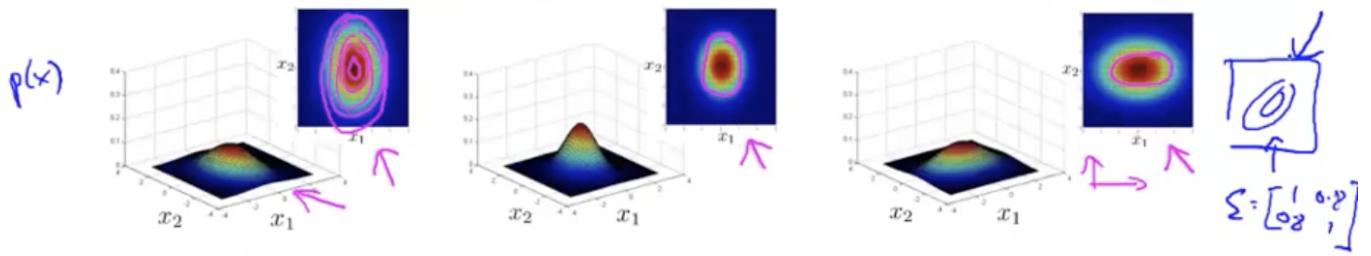
Flag an anomaly if $\underline{p(x) < \varepsilon}$

上图中写出了 $p(x)$ 的计算公式，当我们有一个新的样本 x ，我们就用这个公式来计算 $p(x)$ 的值，如果这个值小于阈值 ε 就把 x 标记为异常

接下来我们对比一下原本的模型和使用多元高斯分布模型的区别：

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

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

上图中我们可以看到， $p(x)$ 原来的模型是 $p(x_1)$ 乘以 $p(x_2)$ 一直乘到 $p(x_n)$ 的积

其实多元高斯模型和原来的模型之间的关系就是，原来的模型其实就是多元高斯分布的一种特例，

即等高线全部都是沿着轴向的（与 x_1 或 x_2 轴平行），这其实就对应了协方差矩阵 Σ 的对角线上没有非零元素存在这一情况。

所以原来的模型其实和多元高斯分布一样，只是多了一个约束，这个约束就是协方差矩阵 Σ 必须满足非对角线元素为0，具体可以看上图中粉色圈标记出的 σ ，把原有式子中的 σ 代入到下面来，就会得到完全一样的模型

但正是由于等高线沿轴向分布，所以我们无法给不同特征变量之间的相关性建模

其实在实际应用中，原来的模型会使用得更加频繁，而只有当我们想要捕捉到特征变量之间的相关性时，

多元高斯分布才具有优势

而原来的模型一个很大的优势是，运算量更小，换句话说，它更适用于 n 的值非常大，也就是特征变量很多的情况

→ 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) $n=10,000, m=100,000$

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

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

$$\Sigma^{-1}$$

Computationally more expensive

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

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

$$m \geq 10n$$

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

上图总结两种模型的优缺点和能对应的情况，注意由于多元高斯分布的公式中要对协方差矩阵 Σ 求逆，

那么就要求 Σ 首先必须要有逆，也就是样本数量 m 必须大于特征数量 n ，而在实际应用中，根据一些经验法则，只有当 m 大于 10 倍的 n 时，我们才使用多元高斯分布。

这里还有一个实际上可能不怎么会遇到的技术细节，就是当我们发现协方差矩阵 Σ 不可逆时，如果并非 m 小于 n ，那么有可能我们有一些冗余特征变量，例如 $x_3 = x_4 + x_5$ 中的 x_3 不包含任何额外信息，就说 x_3 是冗余的