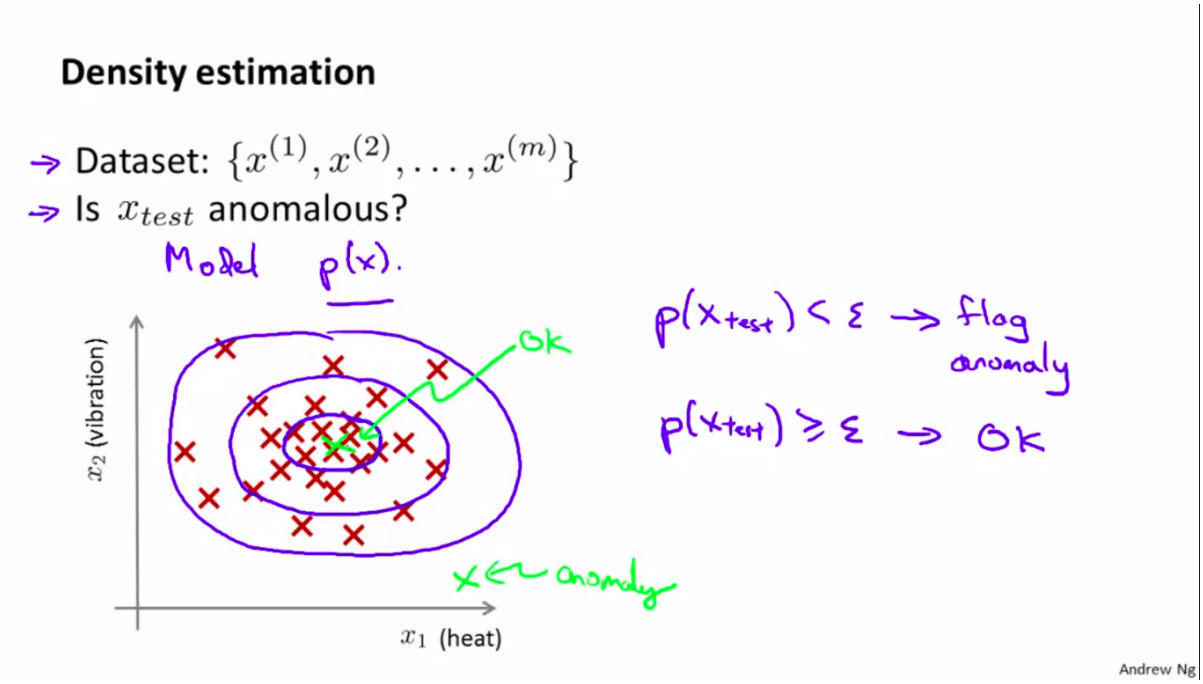
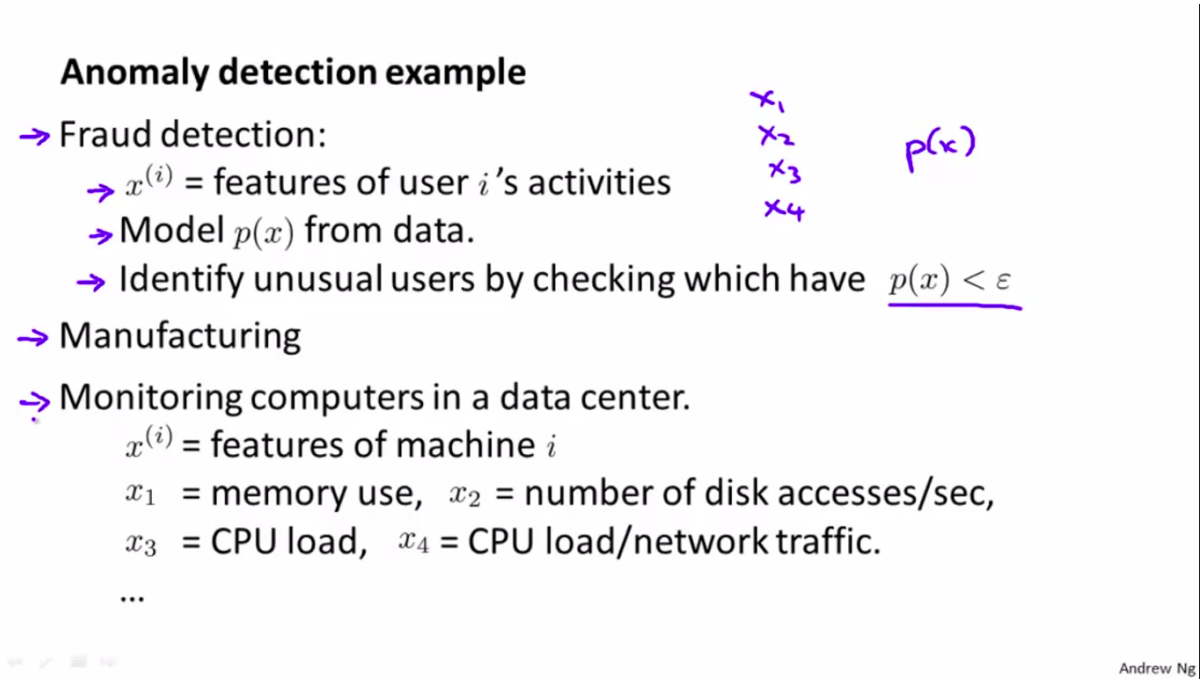
Week 9 - Lecture 1

Density Estimation

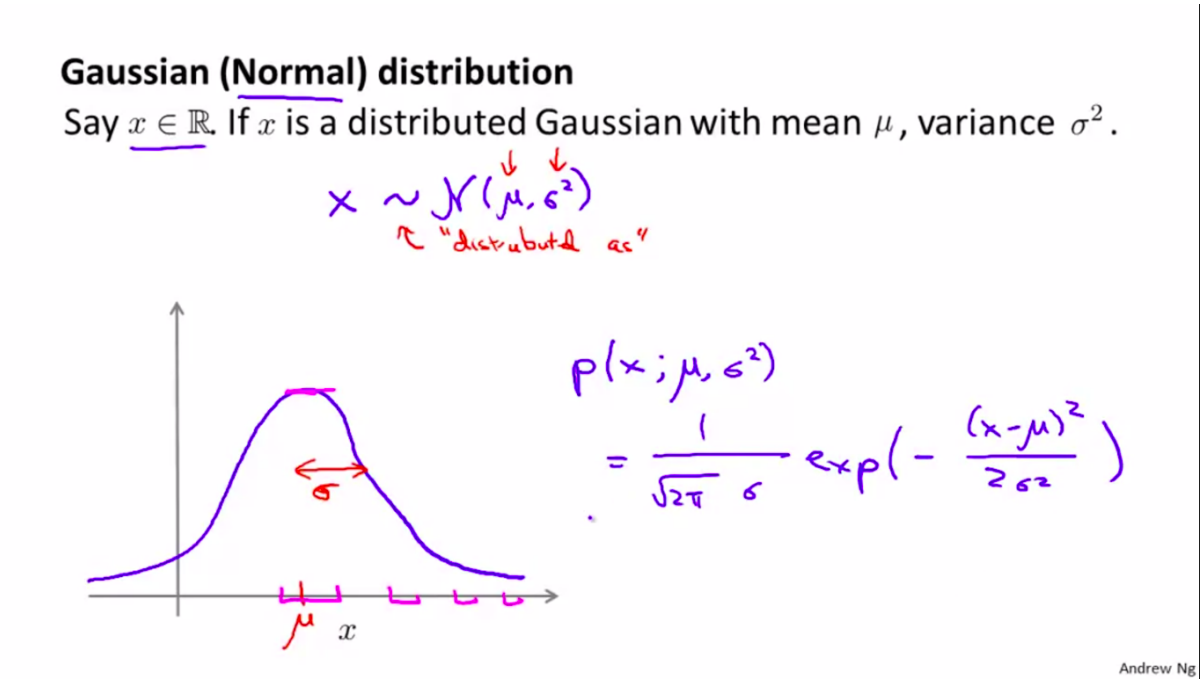
Problem Motivation

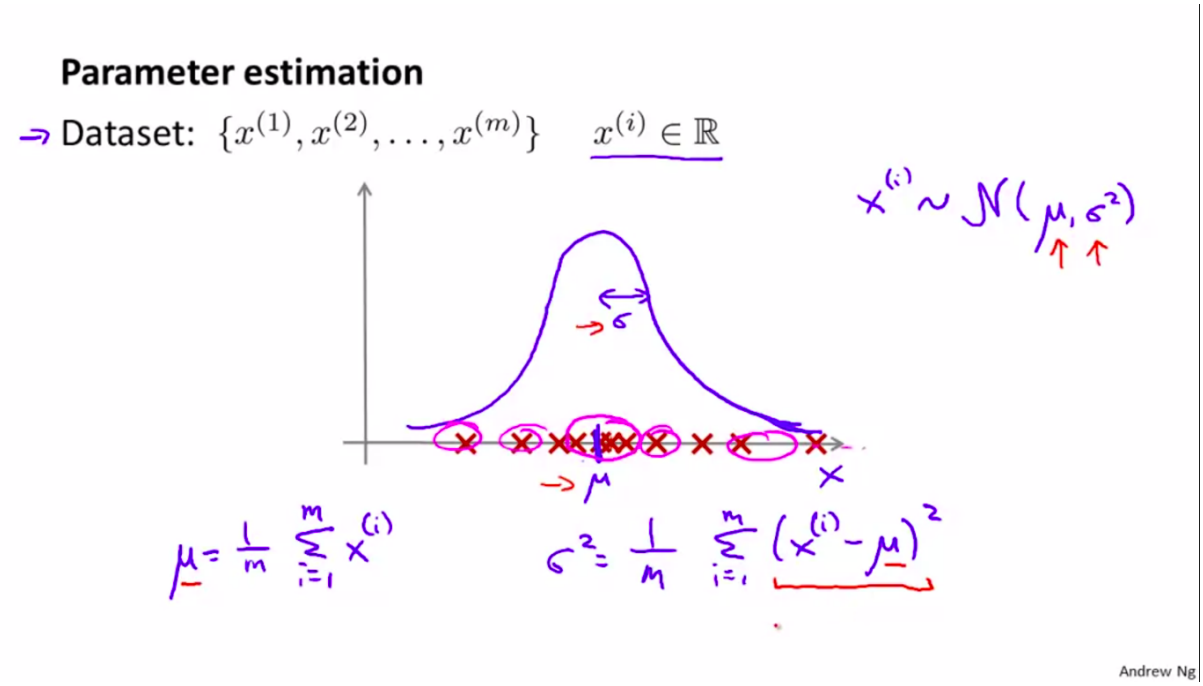
Examples of anomaly detection.





Gaussian Distribution





Note: in the figure above, the parameters formulas corresponds to the Maximum Likelihood Estimators for μ and σ^2 (you may remember that the variance estimator is biased. To make it unbiased, replace $1/m$ for $1/(m-1)$). Professor Ng comments that in Machine Learning people tend to use the Max. Likelihood estimators as is pointed out in the figure and that, in practice, it shouldn't make much difference as long as m is sufficiently large.

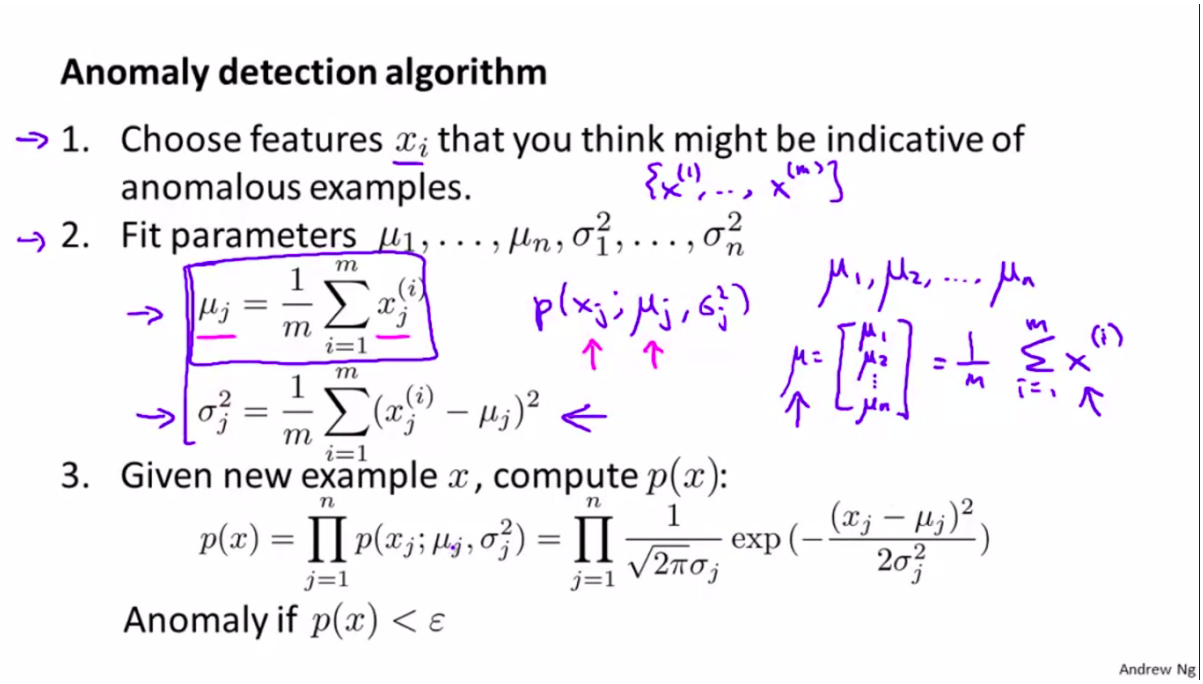
Algorithm

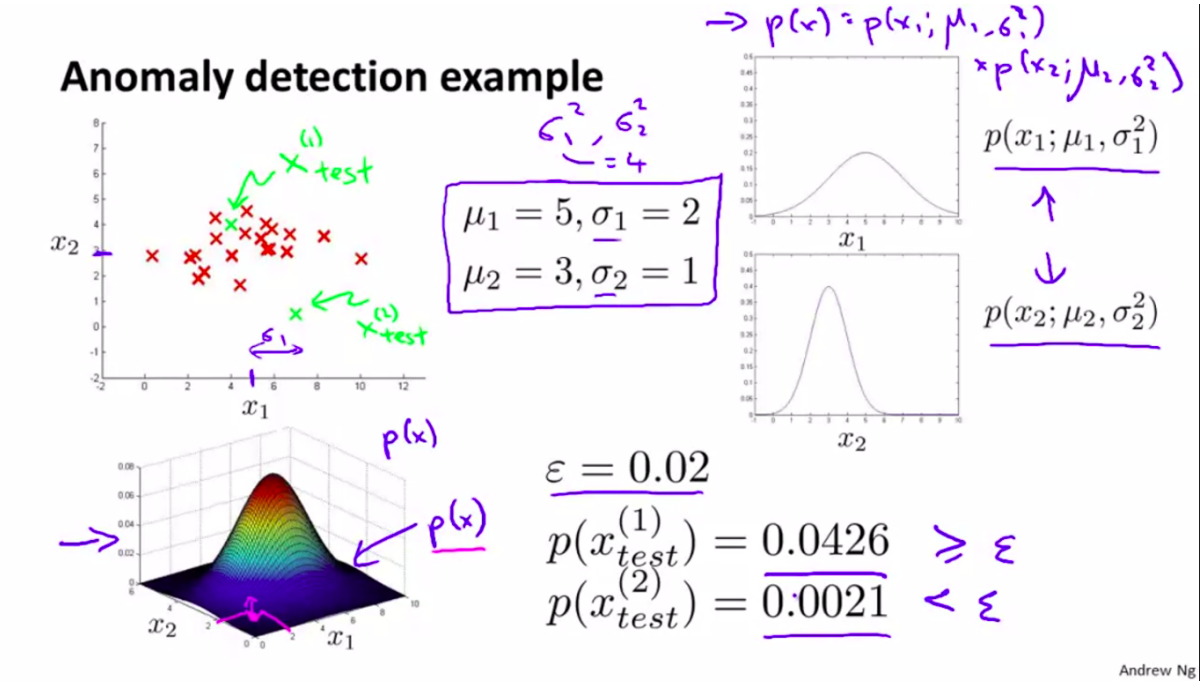
Given a training set $\left\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\right\}$, where each example $x \in \mathbb{R}^n$.

The way we are going to address anomaly detection is by modeling:

$$p(x) = p(x_1;\ \mu_1, \sigma_1^2) \cdot p(x_2;\ \mu_2, \sigma_2^2) \ldots p(x_n;\ \mu_n, \sigma_n^2) = \prod_{i=1}^{n} p(x_j;\ \mu_j, \sigma_j^2) \qquad (1)$$

Note: we're assuming that $x_j \sim \mathcal{N}(\mu_j, \sigma_j)$, for all $j = 1, \ldots, n$. We're also assuming that our features x's are independent.





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: $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ (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: $(x_{test}^{(1)}, y_{test}^{(1)}), \ldots, (x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$

Andrew Ng

Aircraft engines motivating example

→ 10000 good (normal) engines
→ 20 flawed engines (anomalous) 2-50

→ 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:
Training set: 6000 good engines
CV: 4000 good engines ($y = 0$), 10 anomalous ($y = 1$)
Test: 4000 good engines ($y = 0$), 10 anomalous ($y = 1$)

Andrew Ng

Note: commonly, this data will be very skewed (the number of anomalies is much less common than non-anomalies). So, the accuracy metric is not good for evaluating the model's performance in this case (predicting all $y = 0$ would have a good accuracy score).

Algorithm evaluation

- Fit model $p(x)$ on training set $\{x^{(1)}, \ldots, x^{(m)}\}$
- 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

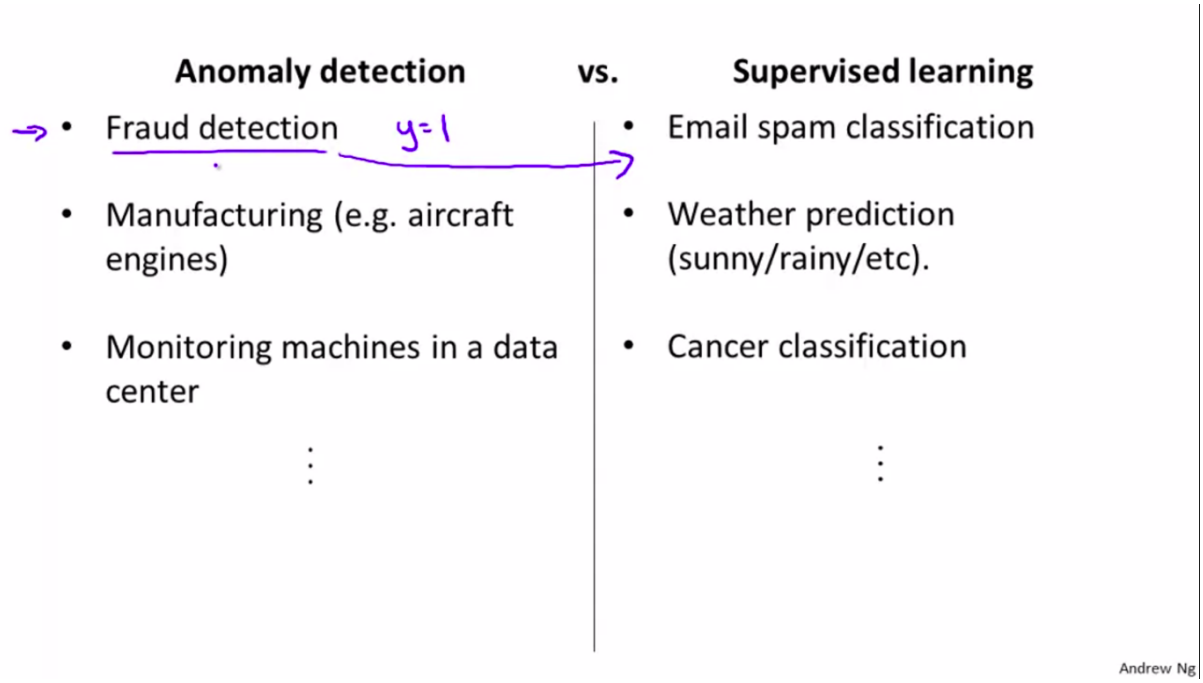
Can also use cross validation set to choose parameter ε

Andrew Ng

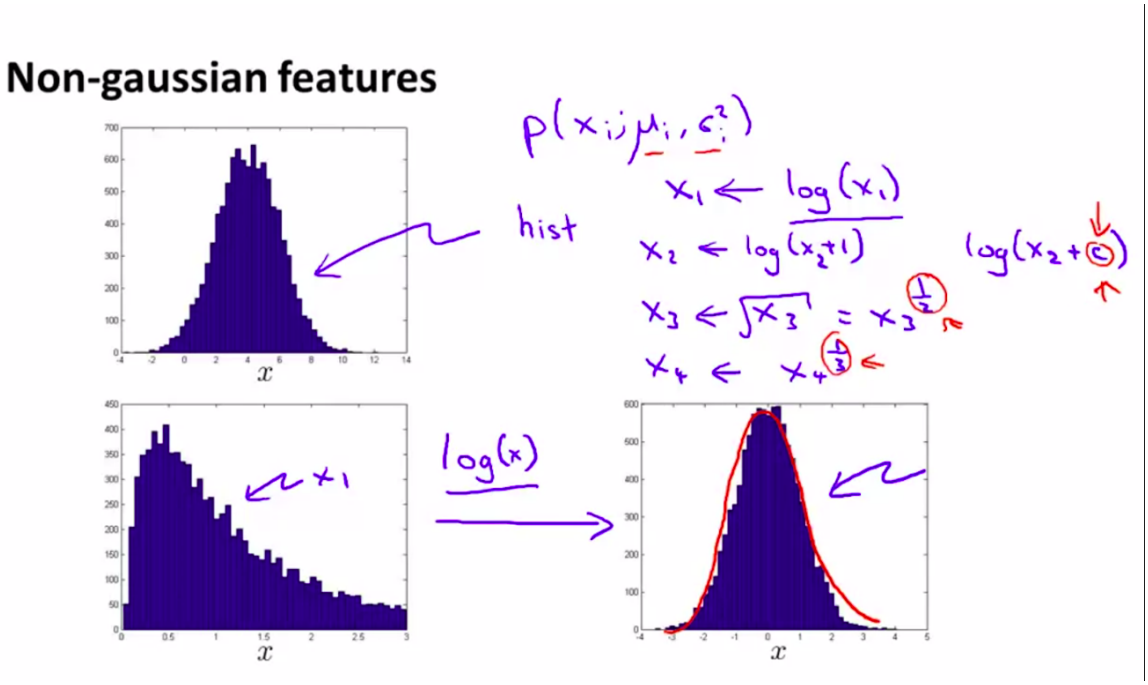
Anomaly Detection vs. Supervised Learning

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.		
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



Choosing What Features to Use

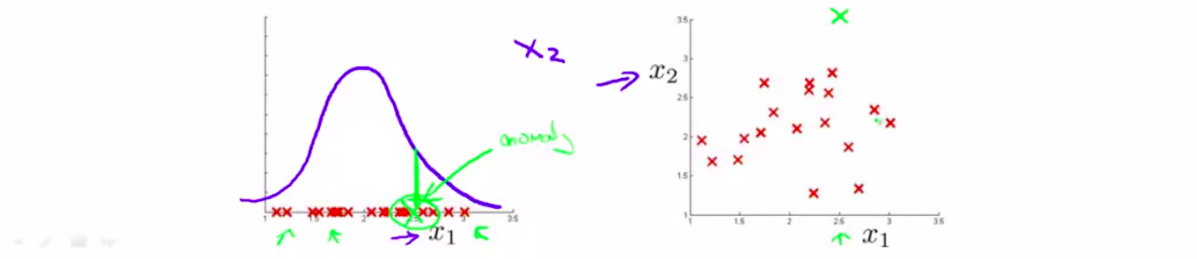


Note: in some cases where the features are non gaussian, the algorithm still works just fine (professor Ng words). However, if a feature doen't look gaussian at all, we can try to find a transformation to make it look like gaussian.

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



What we can do in this case is to look at this anomaly example and try to figure out new features which may help us to dintinguish this "bad" example from the others (like the x_2 variable shown in the figure above. Hopefully, this "bad" example will take on a really unusual value for this new feature).

Example: to try to identify an anomaly in a Data Center, we could try to identify situations in which the CPU load is pretty high, but the network traffic is low (signalling that the CPU might have gotten stuck in a loop). To identify this kind of anomaly, we can come up with new features (x_5 and x_6) like shown in the next figure.

Monitoring computers in a data center

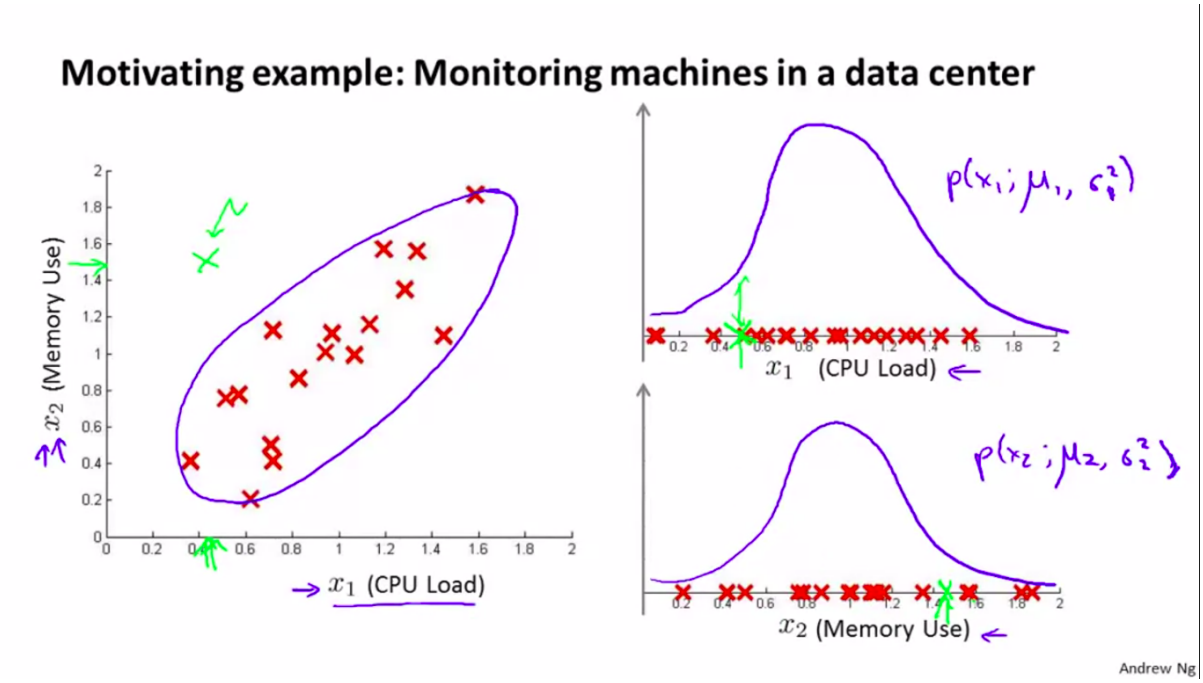
Choose features that might take on unusually large or small values in the event of an anomaly.

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

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

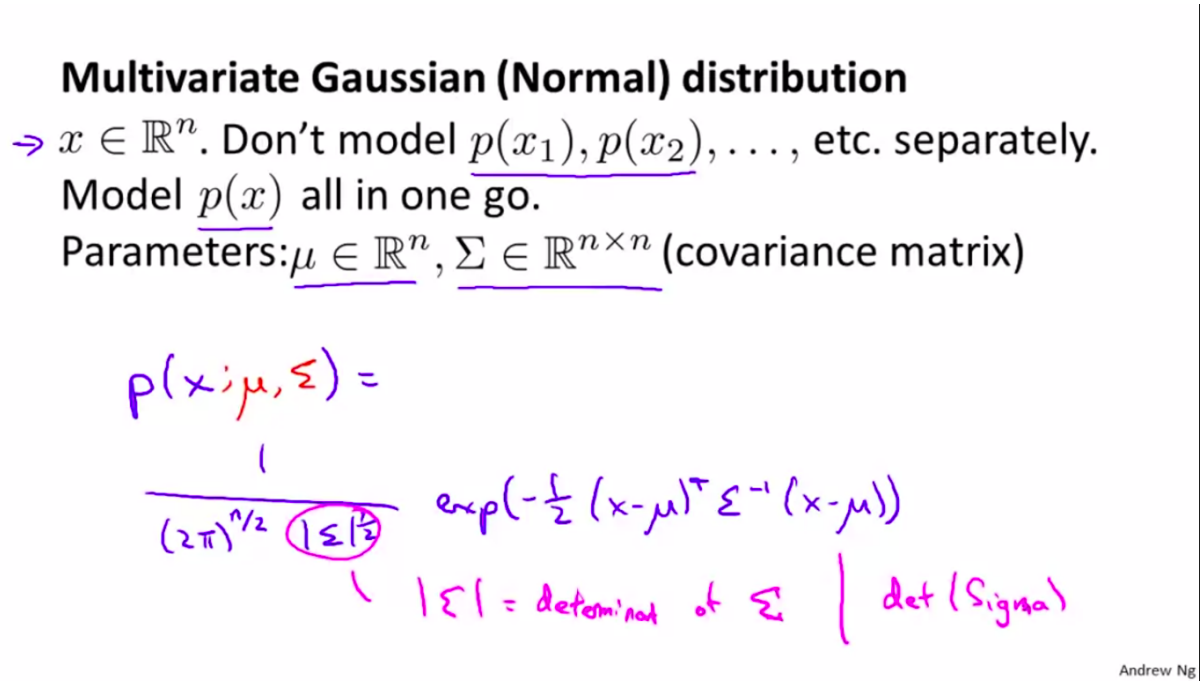
Multivariate Gaussian Distribution

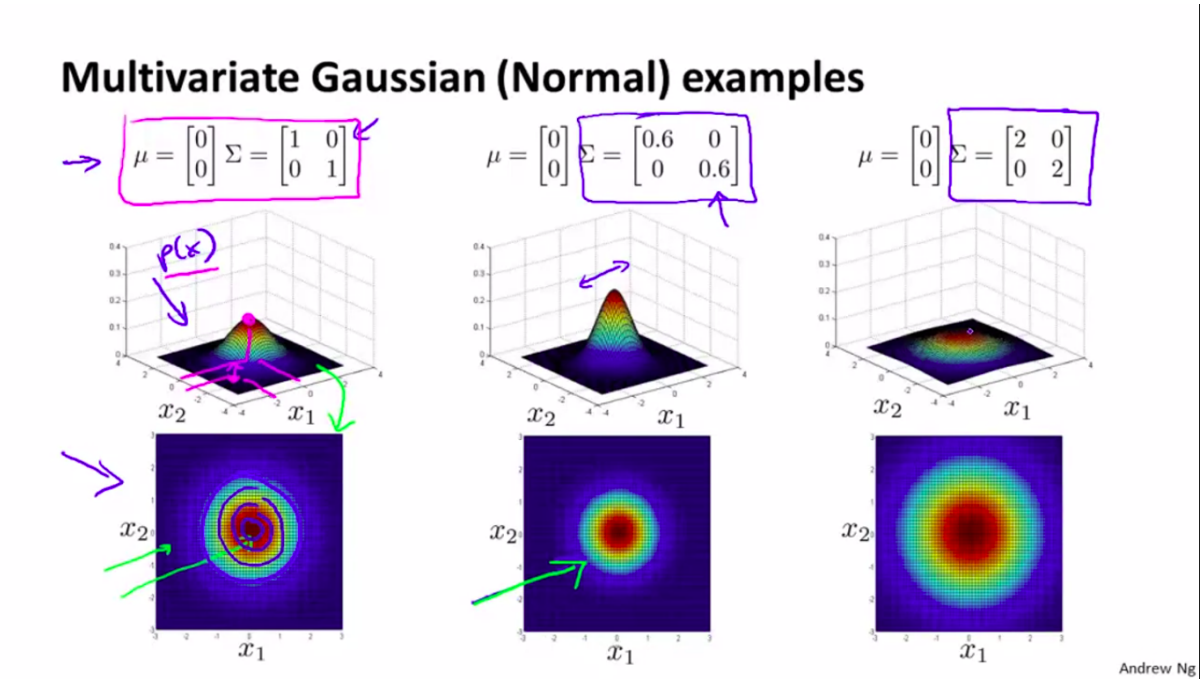
Now we'll start to make an extension to the anomaly detection algorithm we discussed so far.

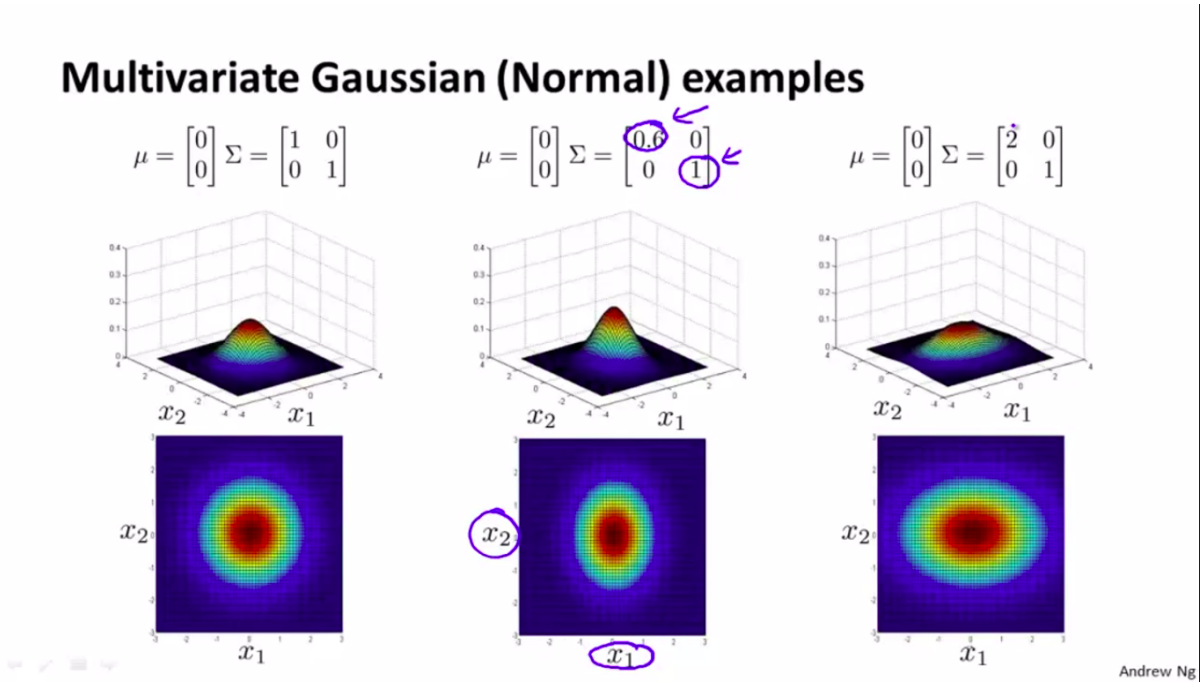


In the example above, it is not easy to distinguish the anomaly (in green) from the rest of normal points. To address this, we are going to develop a Multivariate Gaussian Anomaly Detection Algorithm.

> **Note:** in the previous algorithm, we treated each x_j as independent and identically distributed features and took the joint distribution of them by multiplying all $p(x_j;\ \mu_j, \sigma_j^2)$. In the multivariate Gaussian case, we're going to model all $p(x)$ vector entries in one go.



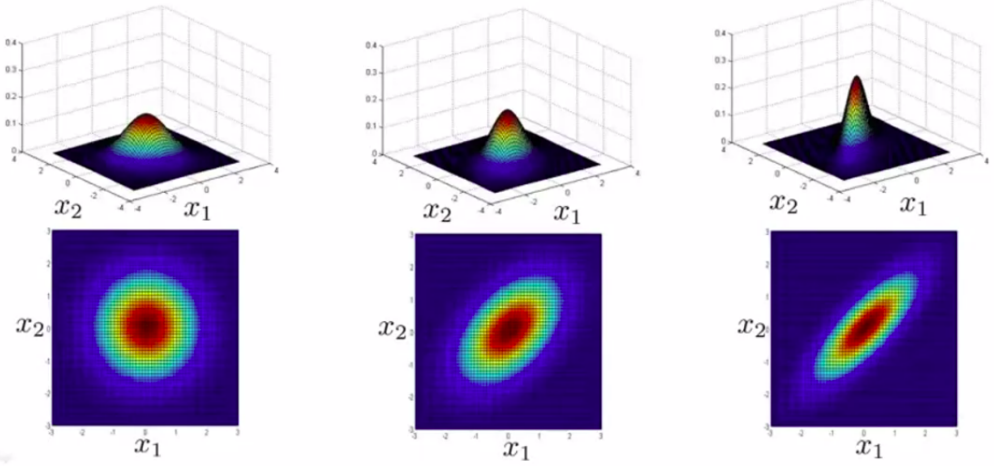




The Multivariate Gaussian Distribution is also good for modelling correlation between the variables:

Multivariate Gaussian (Normal) examples

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



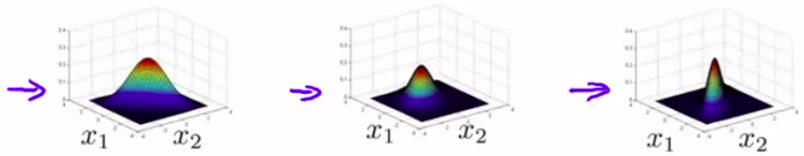
Andrew Ng

Anomaly Detection Using Multivariate Gaussian Distribution

Multivariate Gaussian (Normal) distribution

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

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



Parameter fitting:

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

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

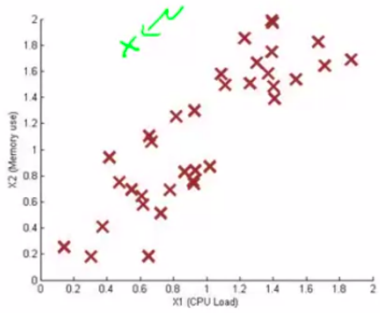
Andrew Ng

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

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

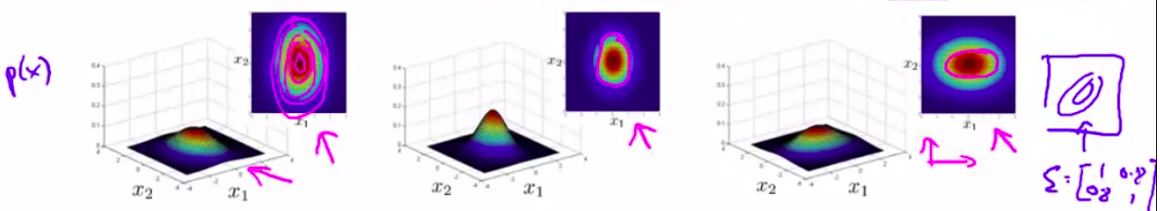
Flag an anomaly if $p(x) < \varepsilon$

Andrew Ng

In fact, the Multivariate Gaussian model is a generalization of the previous model, where the latter corresponds to the former if the contours of the distribution are axis-aligned (i.e., the covariance matrix has all elements outside the main diagonal equal to 0).

Relationship to original model

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



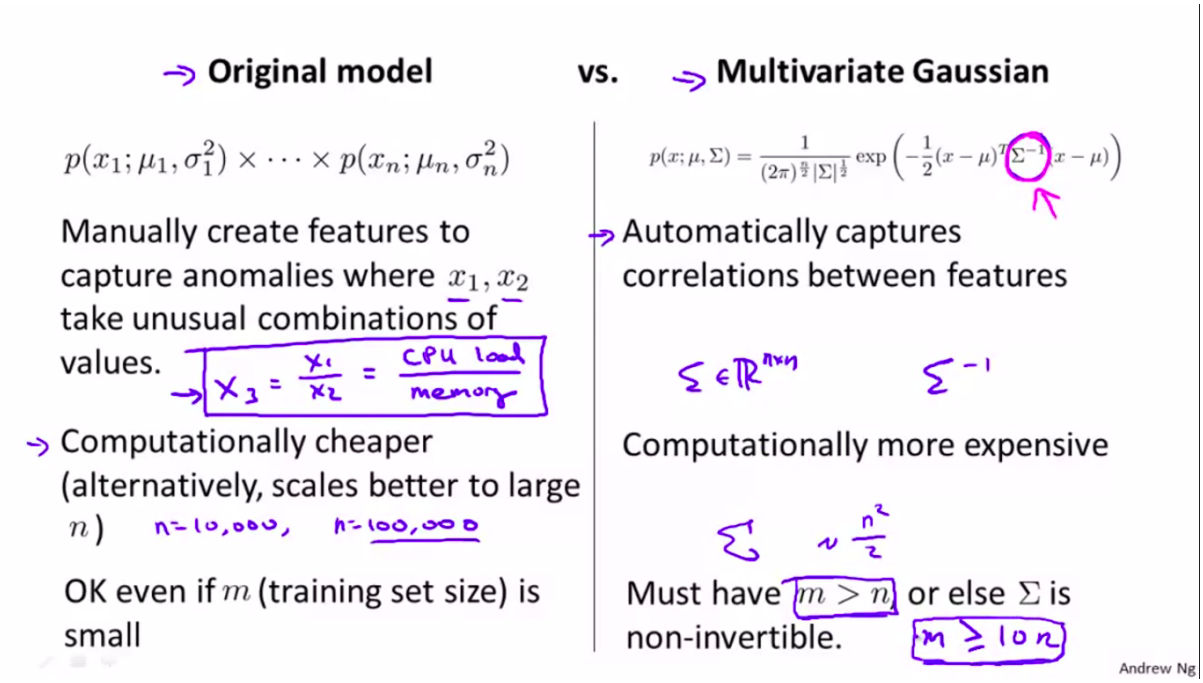
Corresponds to multivariate Gaussian

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

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

Andrew Ng

But **where do we use one model instead of the other?**



References

[1] Machine Learning - Stanford University (https://www.coursera.org/learn/machine-learning)