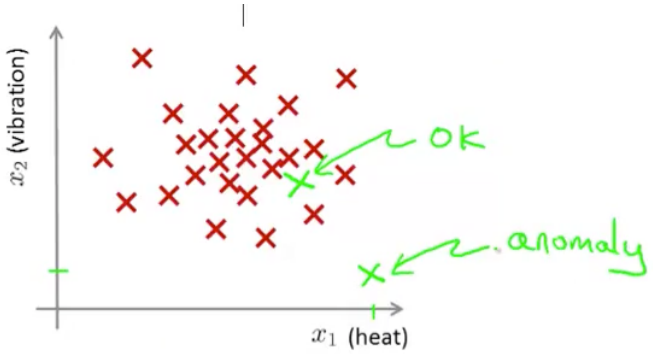
Anomaly Detection

This is one of the most commonly used type of machine learning, which is mixture of unsupervised and supervised problem. As an example, imagine we have a dataset such as heat generated and vibration intensity for jet engine and we have testing and training set, and we would like to know whether we need to do further testing on engine before releasing it or not.



An anomaly detection method is used to see if the new engine is anomalous when compared to the previous engines. If all new engine looks like the red dot, it is ok, but if we get somewhere out of the regions, it is anomaly data point, and we need to send that jet engine back for further testing. In summary, we have an unlabeled dataset, and we would like to build a model p(x) with a probability estimation to find the anomaly. In other word:

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

If $p(x_{test}) \geq \epsilon \rightarrow flag\ ok$

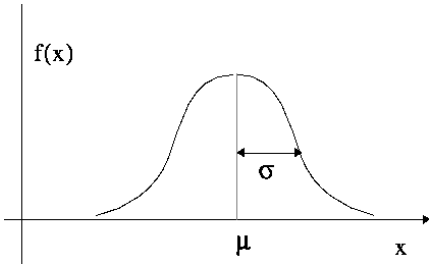
There are so many applications for anomaly detections such as:

- Fraud detections:
  - Features of user i's activity (typing speed, login times...)
  - Model p(x) from data
  - Identify unusual users by checking which have $p(x) < \epsilon$
- Manufacturing: like jet engine we talked about
- Data center
  - Features of machine (memory use, CPU load, number of disk access…)
  - We can identify the anomaly by calculating p(x) and predict which machine is about to go down

Anomaly Detection

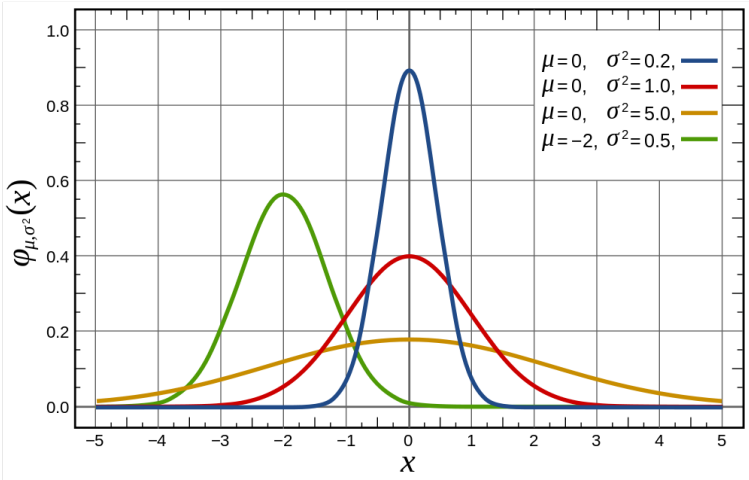
Gaussian Distribution

Let's say $x \in R$, if x is distributed Guassian with mean μ and variance σ^2. We denote it as $x \sim N(\mu, \sigma^2)$, which means x distributed as function N. Gaussian distribution is bell shape. The center of the x is μ and the width of the curve is standard deviation. The Gaussian equation is:



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

We know that area underneath of the plot is equal to1, since it's probability. If $\mu = 0$ and $\mu = 1$, we called it Gaussian distribution. If the std increases, the plot gets narrower and taller. If mean changes, the center of the plot changes to that point



Imagine we have a dataset and we are suspecting whether they come from Gaussian distribution, but we do not know the value of mean and variance. *Parameter Estimation in Gaussian* is as follows:

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^i, \quad \sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(x^i - \mu)^2$$

These parameters are the maximum likelihood estimation values for μ and σ^2. Sometimes, some people might use 1/(m-1), but it doesn't make much difference in calculation of variance.

Anomaly Detection

Density Estimation, Anomaly Detection Algorithm

Imagine there's n unlabeled training set of $\{x^1, x^2, \ldots, x^m\}$ where each of examples $x \in R^n$, we want to figure out the p(x) to know what are high probability features and what are low probability features. We model the probability as follow by assumption of x with Gaussian distribution:

$$p(x)=p\ (x2;\ \mu1\ ,\ \sigma^2 2)\ *\ p(x2;\ \mu2,\ \sigma^2)\ *\ \ldots\ p(xn\ ;\ \mu n\ ,\ \sigma^2 2)$$

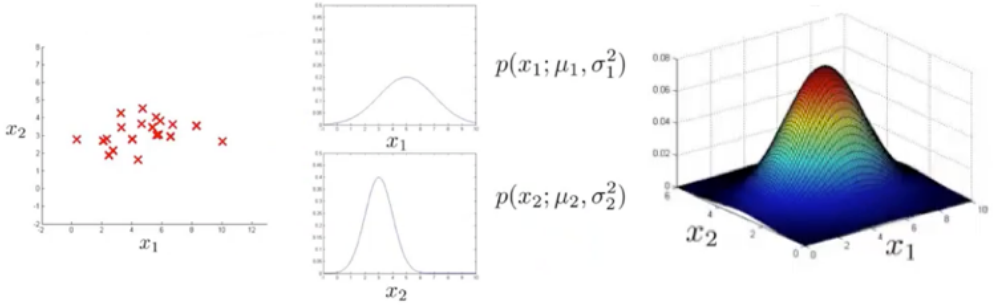
This equation makes an independence assumption for the features, although algorithm works if features are independent or not. We can rewrite the equation as below:

$$p(x) = \Pi\, p(x_j;\ \mu_j, \sigma^2), \text{ for j from 1 to n}$$

Where Capital PI (Π) is the product of a set of values. Sometimes the estimation of distribution is called density estimation. Here is the anomaly algorithm:

1. Choose features x_j that you think might be indicative of anomalous examples.
2. Fit parameters $\mu_1, \ldots \ldots, \mu_n, \sigma_1^2, \ldots . \sigma_n^2$ (equation above)
3. Given new example x, compute p(x)
4. Anomaly if p(x)< threshold

Here's an example, imagine we have a dataset as shown in plot on the left. The dataset on x_1 and x_2 will have the mean and variance shown on the right.



Turns out the p(x) which is product of probability of x_1 and x_2, we get plot shown on the right. For the test dataset, we choose epsilon like 0.02 and if the probability of test set is greater than epsilon, it is not anomaly. In other word, all the points in probability plot that are above the surface are not anomaly and those on the surface are anomaly.

Anomaly Detection

Developing Anomaly Detection System

For anomaly detection, we assume some labeled data of an anomalous and non-anomalous examples (y=0 if normal, y=1 if anomalous). Here are the steps of developing and evaluating anomalous detection:

1. Training set: large collection of unlabeled, normal (not anomalous) dataset with few anomalous
2. Define cross validation set of unlabeled, normal (not anomalous) dataset with few anomalous

Evaluation:

1. Fit model p(x) on training set
2. On a cross validation/test examples x, predict y=1 if $p(x)<\epsilon$ (anomaly), else y=0

   Possible evaluation metrics are:
   - True positive, false negative, false negative, true negative
   - Precision, recall
   - F1 score

Accuracy is not a great way to measure the performance, since we have skewed dataset in anomaly detection. Here is an example. Imagine we have 10,000 normal engines (y=0) and 20 anomalous engines (y=1). A good way is to have 6,000 normal training set, 2,000 normal engines, 10 anomalous cross validation set, and 2,000 normal engines, 10 anomalous in testing set.

How can we pick the epsilon? We try different epsilon and see which one maximizes the value of f1 score.

Anomaly vs supervised learning

Here is the table comparing these two:

Anomaly Detection	Supervised learning
• If we have small number of positive examples, y=1 (0-20 is common for anomalies), and large number of negative, normal examples (y=0) We use positive values only for test and cross validation set, since we have very few of them. We use negative examples for making a model p(x). In this case, since we have few positive examples, it's hard for algorithm to learn them. Examples: fraud detection, monitoring machines in a datacenter…	• Large number of positive and negative example. Examples: cancer classification, weather prediction, email spam classifications…

Anomaly Detection

What Features to Use

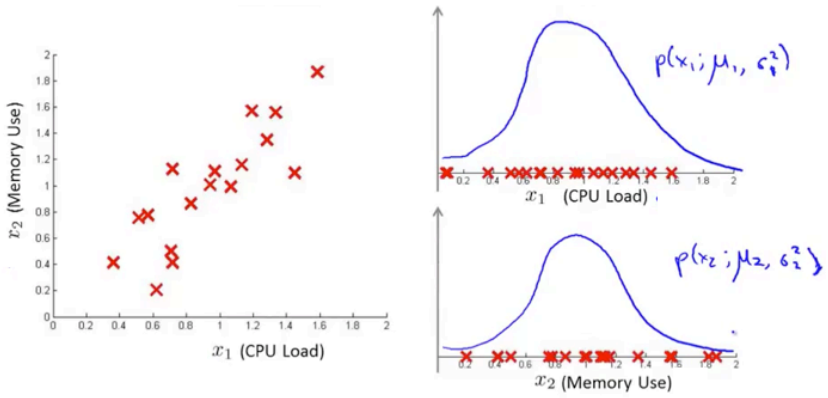
Features we choose has a huge effect on anomaly detection. We used Gaussian distribution to model the features for anomaly algorithm. You may plot the histogram to make sure the data looks vaguely Gaussian. *If it is not looking semi Gaussian, take a log or $x^2 + c$... of that, and it may look Gaussian.* For example, make replace $x_1$with $\log(x_1)$, and $x_2$with x^2 to make the hist Gaussian.

Error Analysis for Anomaly Detection

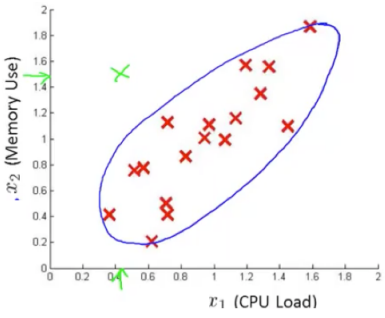
We are hoping to have large value of p(x) for normal example and small value for anomalous examples x. One of the most common problem is when p(x) is comparable for normal and anomalous examples. In this case, for example if an anomaly detects as normal, we take a look at what went wrong and we may change a feature x to make the regions clearer to decide between. *Choose features that might take on unusually large or small values in the events of an anomaly*. For example, if we have features like CPU loads, number of disks, memory use, traffic network, we may create a new feature (CPU load) ^{2}/ (network traffic) which this value will be large if we happen to have a infinite loop in a network.

Multivariate Gaussian Distribution

This helps catch anomalies we couldn't capture in previous anomaly detection algorithm. In previous example, we talked about features in data center problem. Let's consider two features of memory use and CPU load, shown on the left. Plotting x_1, x_2 gives Gaussian shape as shown on the right plot below:



After anomaly detection, if a data points shown as green dot appears, we may consider it as anomaly since it's away from the region of normal. That green dot might be a point related to p(x) with similar values to normal values in p(x_1) plot and p(x_2) plot, and this case, the algorithm fails to predict correctly. In order to fix this problem, we use multivariate Gaussian(normal) distribution. In



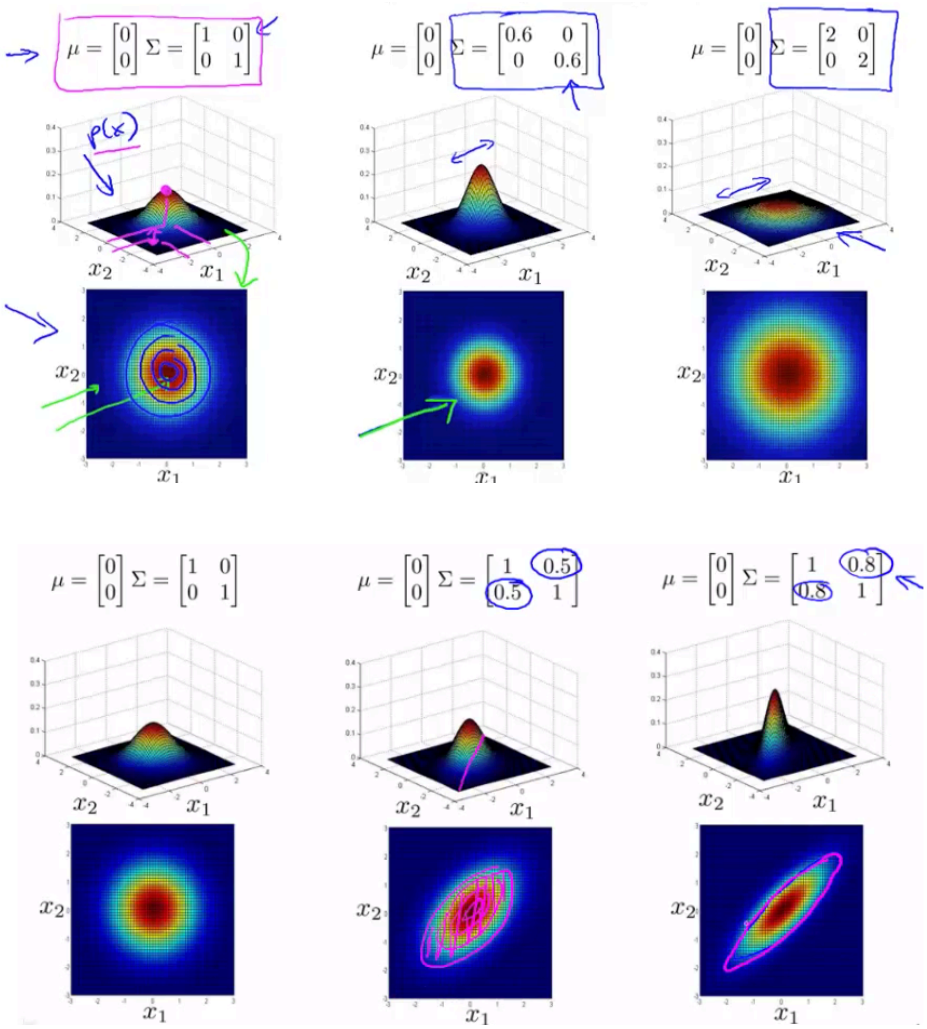
Anomaly Detection

this model for do not model p(x_1) plot and p(x_2) separately and we model p(x) all in one go.

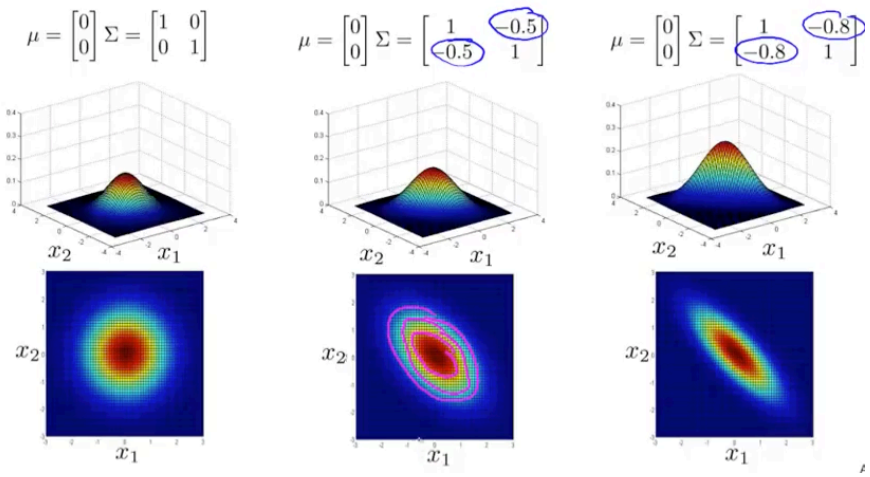
Here is the algorithm for multivariate Gaussian(normal) distribution: (Imagine $\mu \in R^n, \Sigma \in R^{n*n}$.

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

Where $|\Sigma|^{1/2}$is called determinate of Σ. It can be calculated with |det (sigma)|. Let's take a look at example as shown on the right. The peak of the figure, shows the value of the p(x) at each point. Most of probability are near 0 and 0 and probability lowers down as we go further from 0. As we decrease the value of variance, the plot gets narrower and taller. Below is example of what happens by changing the variance values (Gaussian Distribution changes):



Anomaly Detection



These also gives correlation between features as well, for example when variance are negative in diagonal, x_1, x_2 have negative correlations. Chang value of μ, will shift the center of distribution.

Anomaly Detection using Multivariate Gaussian Distribution

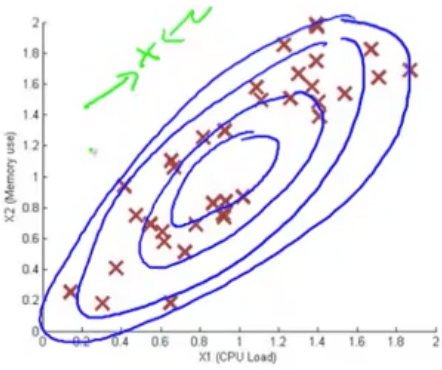
The multivariate Gaussian has two parameters of $\in R^n, \Sigma \in R^{n*n}$. How can we parameter fitting by given the training dataset of x? That's easy, we fit the values in the equations:

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

So Here are the steps for Multivariate Gaussian:

1. Fit the training set into mean and variance equation above
2. Given a new sample x, compute the p(x)
3. Flag an anomaly if p(x)< threshold.

Applying this algorithm will give high probability at the center of dataset, less as we go further and correctly predict x if it's anomaly.



The original model is exactly similar to the equation of multivariate Gaussian distribution, and the only difference is that multivariate Gaussian distribution gives us angled (like 45 degrees in our example) Gaussian distribution shown similar to the plot on the right. In other word, multivariate Gaussian distribution is exactly same

equation as our original ones if variance has variance of features only diagonally. Here are the difference between these two model:

Original model	Multivariate Gaussian Distribution
$p(x)=p(x2;\mu1,\sigma^2 2) * p(x2;\mu2,\sigma^2) * \ldots p(xn;\mu n,\sigma^2 2)$	$p(x;\mu,\Sigma) = \frac{1}{(2\Pi)^{n/2}\lvert\Sigma\rvert^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$
• Commonly used • Manually creates features to capture anomalies where x_1, x_2 takes unusual combinations of values, create extra features if the necessary. • Computationally cheaper (alternatively, scale better to large n) • Ok even if m (training set) is small	• Automatically captures correlations between features • Computationally expensive (since we have to calculate inverse) • Must have m>n, or else Σ is not-invertible (usually m>10 n in practice), or if we have redundant features (two features that capture similar information, highly correlated)