# Anomaly Detection

e.g. aircraft engine features

$x_1$ = heat generated

$x_2$ = vibration intensity

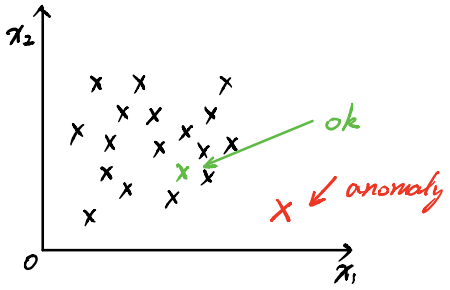
.....

dataset: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$

new engine : $x_{test}$

$p(x) \geq \varepsilon$ ok

$p(x) < \varepsilon$ anomaly



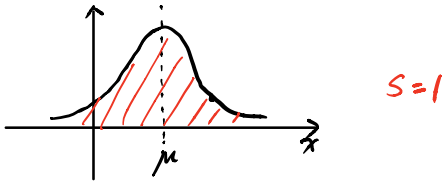

anomaly detection example

- fraud detection
- manufacturing (example above)
- monitor computers in data center

# Gaussian distribution

$x \in R$, if $x$ is distributed Gaussian with mean $\mu$, variance $\sigma^2$

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




$S = 1$

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

parameter estimation:

$\{x^{(1)}, x^{(2)}, \cdots, x^{(m)}\}$ , $x^{(i)} \in R$

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

Square error

anomaly detection algorithm:

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

training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$ $x^{(i)} \in \mathbb{R}^n$

$$x_j \in N(\mu_j, \sigma_j^2) \quad j = 1, 2, \cdots, n$$

2. fit parameters $\mu_1, \cdots, \mu_n, \sigma_1^2, \cdots, \sigma_n^2$

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)} \qquad j = 1, 2, \cdots, n$$

$$\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}(x_j^{(i)}-\mu_j)^2 \qquad j = 1, 2, \cdots, n$$

3. give new example $x$, compute $p(x)$

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

if $p(x) < \varepsilon \Leftrightarrow$ anomaly

algorithm evaluation ( how to choose $\varepsilon$ ? )

fit model $p(x)$ on training set $\{x^{(1)}, x^{(2)}, \cdots, x^{(m)}\}$

on cross validation/test example $x$, predict

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

evaluation metrics:

- true positive, false positive, true negative, false negative
- precision / recall
- $F_1$ - score

anomaly detection

mainly normal cases, few anomaly cases.

many different type of cases hard for algorithm to learn.

- fraud detection
- manufacturing
- monitoring machines

⋮

vs

supervised learning

large number of positive and negative cases

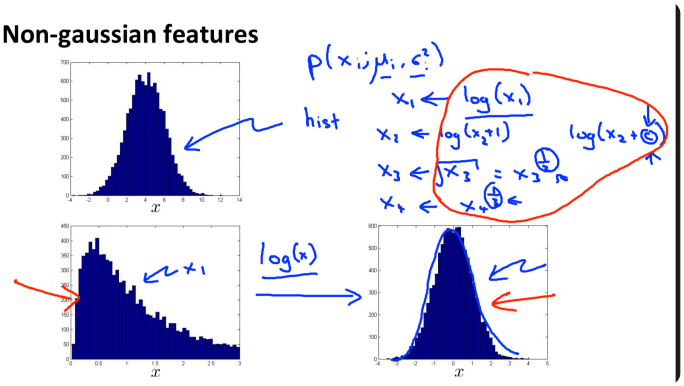
enough positive cases for algorithms to learn.

- spam email classification
- weather prediction
- cancer classification

⋮

choosing what features to use.

check each dimension of data manully to see if the feature if gaussion or not. If they are non-gaussian, we need to make them gaussian by using: $x \leftarrow \log(x+c)$ or $x \leftarrow x^a$



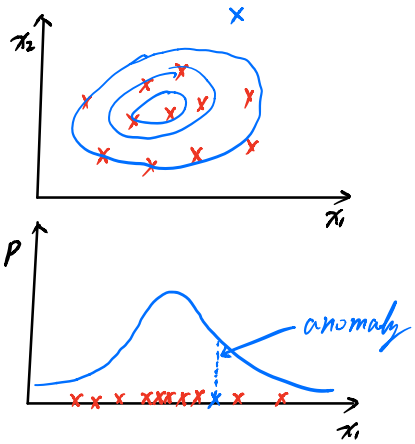
error analysis for anomaly detection

want $p(x)$ large for normal examples $x$

want $p(x)$ small for anomalous examples $x$

most common problem:

$p(x)$ is comparable (say, both large) for normal and anomaly examples.

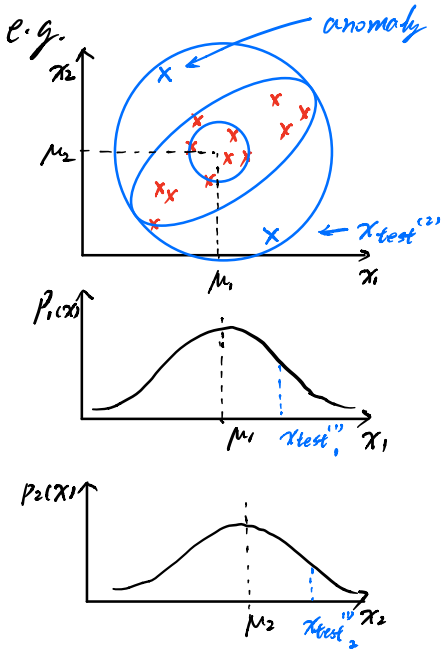



$x_1$ = memory use of computer

$x_2$ = CPU load

we now need to create a new feature

$x_3 = \frac{x_1}{x_2}$ or $\frac{x_1^2}{x_2}$ ...




$p(x_{test}^{(1)}) = p_1(x_{test,1}^{(1)}) \times p_2(x_{test,2}^{(1)}) < \varepsilon$

× predict normal but actual anomaly

for the boundary is ○ rather than ○

can add new feature $x_3 = \frac{x_1}{x_2}$

or use "multivariate gaussian distribution"

## Mutivariate Gaussian Distribution

$x \in \mathbb{R}^n$, model $p(x)$ all in one instead of modelling $p(x_1), p(x_2) \dots$ separately

parameters: $\mu \in \mathbb{R}^n$, $\Sigma \in \mathbb{R}^{n \times n}$ (convariance matrix)

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

parameter fitting:

giving training set $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$

$$\mu = \frac{1}{m} \sum_{i=1}^{m} x^{(i)} \in \mathbb{R}^n$$

$$\Sigma = \frac{1}{m} \sum_{i=1}^{m} (x^{(i)} - \mu)(x^{(i)} - \mu)^T \in \mathbb{R}^{n \times n}$$

relation between the origin model and new model:

origin $p(x) = p(x_1; \mu_1, \sigma_1^2) \times p(x_2; \mu_2, \sigma_2^2) \times \dots \times p(x_n; \mu_n, \sigma_n^2)$

in new model:

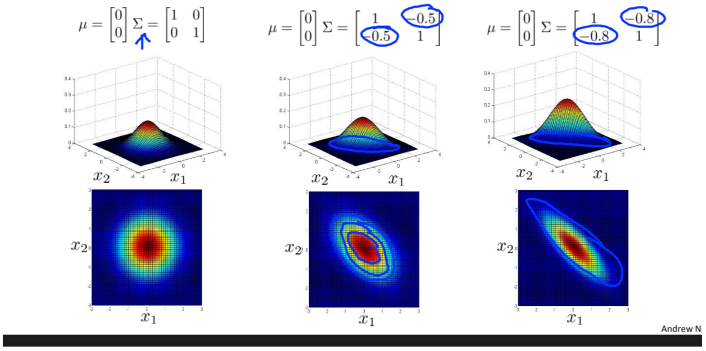
$$\Sigma = \frac{1}{m} \sum_{i=1}^{m} (x^{(i)} - \mu)(x^{(i)} - \mu)^T = \begin{bmatrix} \sigma_1^2 & & \\ & \sigma_2^2 & \\ & & \ddots \\ & & & \sigma_n^2 \end{bmatrix}_{n \times n}$$

| origin model | mutivariate Gaussian distribution |
| --- | --- |
| manually create features to capture anomalies where $x_1, x_2$ take unusual combination of values. $x_3 \leftarrow \frac{x_1}{x_2}$ | automatically captures correlations between features. |
| computationally cheaper, scale better to large ok if $m$ is small | computationally more expensive must have $m > n$ ($m > 10n$) $\Sigma$ be invertible |

when the graph like ○, ⬭, ⬮ , use origin model directly

when the graph like ◯, ◯ , use multivariate gauss distribution

**Multivariate Gaussian (Normal) examples**




**Multivariate Gaussian (Normal) examples**