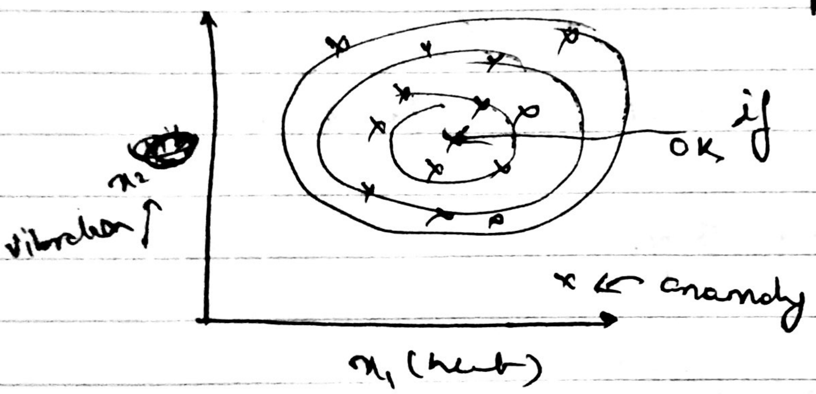
→ only used to speed up algorithm.

## ★ ANAMOLY DETECTION

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

Is $x_{test}$ anomalous?

Model $p(x)$.




if $p(x_{test}) < \epsilon$ → flag anomaly

$p(x_{test}) \geq \epsilon$ → OK.

### Application

→ Fraud detection.

→ Manufacturing

→ Monitoring computers in a data center.

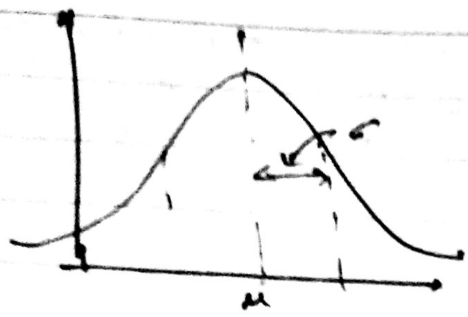
## GAUSSIAN DISTRIBUTION / NORMAL DISTRIBUTION

$x \in \mathbb{R}$ If $x$ is a distributed gaussian with mean $\mu$, variance $\sigma^2$. ($\sigma$ - standard deviation)

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

"distributed as"

independence assumptions




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

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

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

Unlabelled dataset
Training set = $\{x^{(1)}, \dots, x^{(m)}\}$
Each example is $x \in R^n$.

$x_1 \sim N(\mu_1, \sigma_1^2)$
$x_2 \sim N(\mu_2, \sigma_2^2)$

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

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

Algorithm

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

2) Fit parameters $\mu_1, \dots \mu_n$,
$\sigma_1^2, \dots \sigma_n^2$,

$$\mu_j = \frac{1}{m} \sum_{i=1}^{m} x_j^{(i)}$$ ..... [vectorized implementation. $\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_n \end{bmatrix} = \frac{1}{m} \sum_{i=1}^{m} x^{(i)}$ ]

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

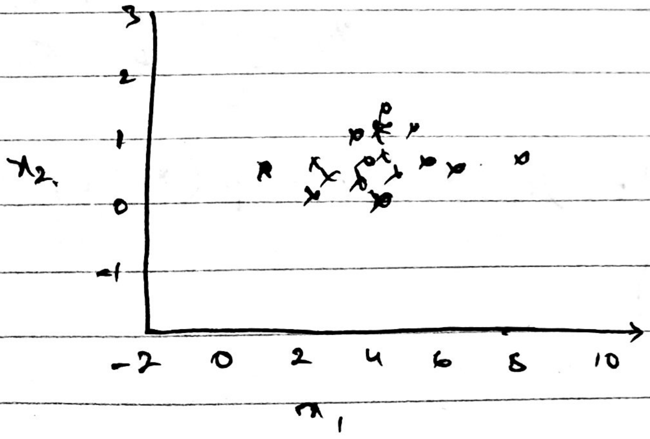
3). Given new example $x$, compute $p(x)$:

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

$$= \prod_{j=1}^{n} \frac{1}{(\sqrt{2\pi})\sigma_j} \exp\left[-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}\right]$$

Anomaly if $p(x) < \varepsilon$.

Eg




$\mu_1 = 5 \quad \sigma_1 = 2$

$\mu_2 = 0.5 \quad \sigma_2 = 0.5$

**Real No. evaluation** – When developing learning algorithm, ~~knowing~~ a certain feature is added or excluded.

**Algo**

→ fit model $p(x)$ on training set $\{x^{(1)}, x^{(2)}, \dots x^{(m)}\}$
On a cross validation / test set, predict

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

# Similar to classification however, it's skewed since $y=0$ (normal engine) is way more than $y=1$ (anomalous engine).

Hence, we can use evaluation metrics:
- True positive, false positive, false negative, true negative.
- Precision / Recall
- $F_1$ - score

$\epsilon$ is evaluated on cross-validation sets.

~~Supervised Learning v/s Anomaly detection~~

Choosing what features to use

1) plot data to see if your data is in gaussian or not

→ If data is not gaussian, take $\log(x)$
or use $\log(x + c)$ ← constant

hist(x^0.05);

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

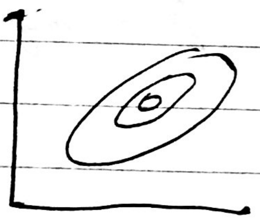
## Multivariate Gaussian distribution

→ $x \in \mathbb{R}^n$. Don't model $p(x_1)$, $p(x_2)$, ...., etc.
Parameters : $\mu \in \mathbb{R}^n$, $\Sigma \in \mathbb{R}^{n \times n}$ (covariance matrix)

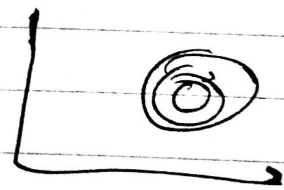
$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

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

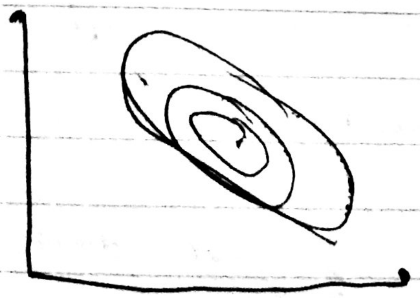
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $\Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$



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



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



$\mu \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ → if changed, the position is shifted

Anomaly detection using multivariate gaussian distribution

Parameters - $\mu, \Sigma$ ; $\mu \in R^n$ $\Sigma \in R^{n \times n}$

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

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

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

---

## RECOMMENDER SYSTEMS

recommends projects or items

$n_u$ = no. of users

$n_m$ = no. of movies

$r(i,j) = 1$ if $j$ has rated movie $i$

$y(i,j)$ = rating given by user $j$ to movie $i$ (defined iff $r(i,j) = 1$)

" Recommender ~~system~~ should predict the movie ratings that are not ~~filled~~ given by users & also predict what else can the user watch.