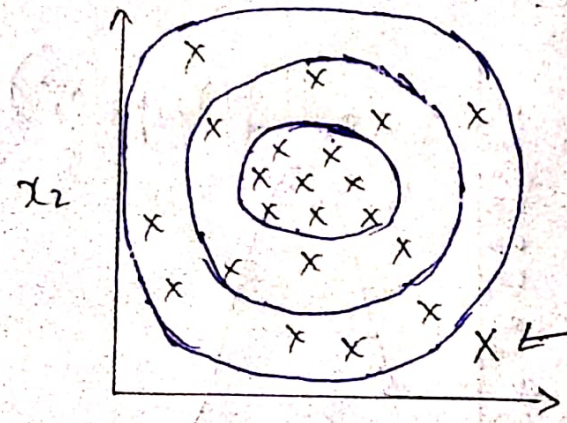
ANOMALY DETECTION

Definition :-

Anomaly detection is a step in data mining that identifies data points, events, and/or observations that deviate from the dataset's normal behaviour.

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

Is x_{test} anomalous ?



$p(x_{test}) < \epsilon \rightarrow$ Anomaly

$p(x_{test}) \geq \epsilon \rightarrow$ Normal

fig1 :- Model $p(x)$

Eg :- Fraud detection :-

→ $x^{(i)}$ = features of user i's activities

→ Model $p(x)$ from data

→ Identify unusual users by checking which have $p(x) < \epsilon$

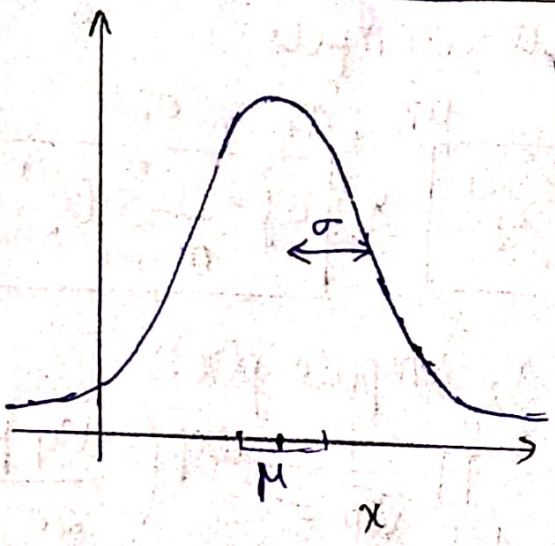
GAUSSIAN DISTRIBUTION

Definition :-

Gaussian distribution or Normal Distribution is a bell shaped curve, and it is assumed that during any measurement values will follow a normal distribution with equal number of measurements above and below the mean value.

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

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



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

where μ = mean
σ = standard deviation
σ^2 = variance

Parameter Estimation :-

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

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

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

↑ mean

↑ variance

③ # ANOMALY DETECTION ALGORITHM

Density estimation

Given Training set : $\{x^{(1)}, x^{(2)}, \ldots\ldots x^{(m)}\}$
where each example is $x \in R^n$

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

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

$\sum_{i=1}^{n} i = 1+2+3\ldots+n$

$\prod_{i=1}^{n} i = 1\times2\times3\times\ldots\times n$

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

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

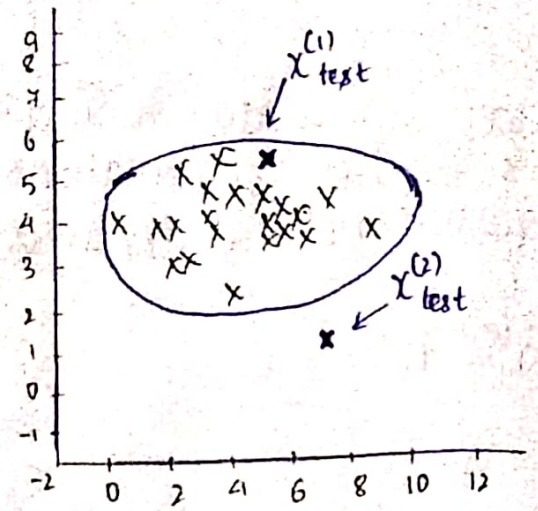
where $\boxed{\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)}}$ & $\boxed{\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}(x_j^{(i)} - \mu_j)^2}$

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

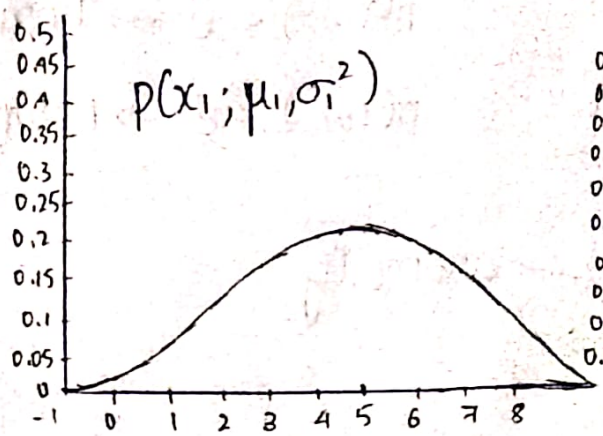
$$p(x) = \prod_{j=1}^{n} p(x_j;\mu_j,\sigma_j^2) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{(x_j-\mu_j)^2}{2\sigma_j^2}\right)$$

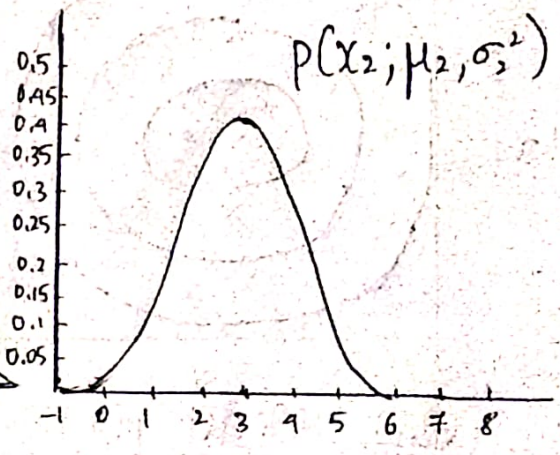
Anomaly if $p(x) < \epsilon$

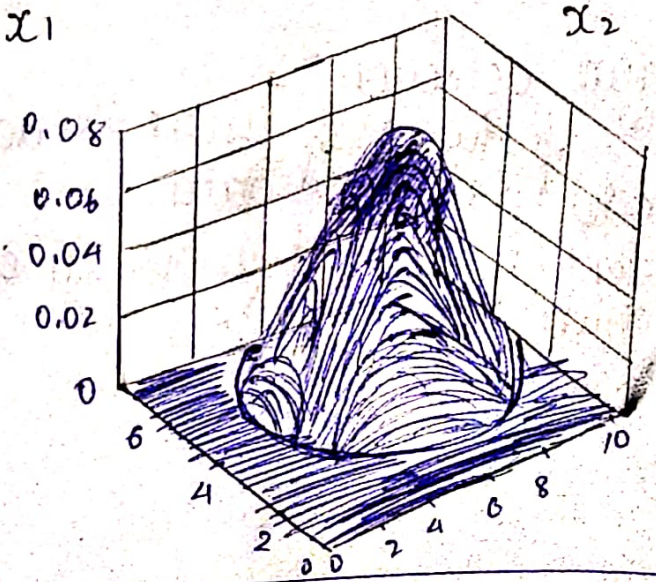
Example (4)



$$\mu_1 = 5, \sigma_1 = 2$$
$$\mu_2 = 3, \sigma_2 = 1$$







$$p(x) = p(x_1; \mu_1, \sigma_1^2) \times p(x_2; \mu_2, \sigma_2^2)$$

If $\epsilon = 0.02$ then $p(x^{(1)}_{test}) = 0.0426 \geq \epsilon$ (Normal)

$p(x^{(2)}_{test}) = 0.0021 < \epsilon$ (Anomaly)

(5)

Developing and Evaluating Anomaly Detection Algorithm

When developing a learning algorithm (choosing features) making decisions is much easier if we have a way of evaluating our algorithm.

→ Assume we have some labelled data of anomalous and non-anomalous examples
i.e $y=0$ if normal and $y=1$ if anomalous

→ Training set : $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ contains all normal / non-anomalous examples

→ Cross validation set : $(x_{cv}^{(1)}, y_{cv}^{(1)}), \ldots, (x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})})$
contains both normal and anomalous examples

→ Test set : $(x_{test}^{(1)}, y_{test}^{(1)}), \ldots, (x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$
contains both normal and anomalous examples

Example:

→ Consider 10000 normal transactions and 20 anomalous transactions

→ Training set : 6000 normal transactions

→ CV set : 2000 normal transactions ($y=0$)
10 anomalous transactions ($y=1$)

→ Test set : 2000 normal transactions ($y=0$)
10 anomalous transactions ($y=1$)

⑥ Algorithm Evaluation

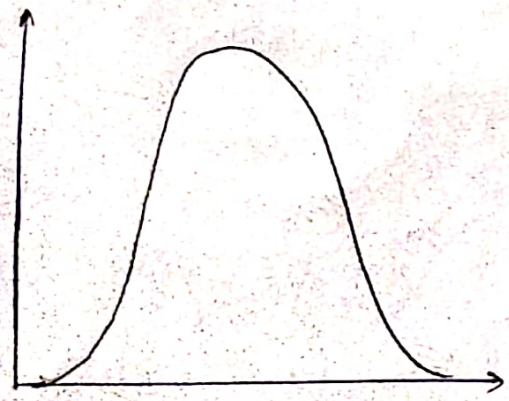
→ Fit model $p(x)$ on training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$

→ On cross validation / test example x, predict

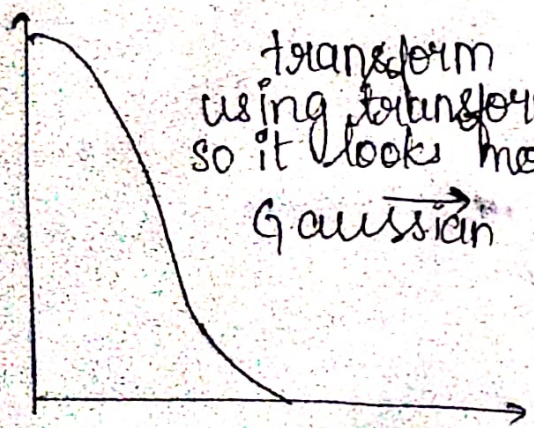
$$y = \begin{cases} 1 & \text{; if } p(x) < \varepsilon \text{ (anomaly)} \\ 0 & \text{; if } p(x) \geq \varepsilon \text{ (normal)} \end{cases}$$

Anomaly detection	vs	Supervised Learning
→ Very small no. of positive examples ($y = 1$)		Large no. of positive and negative examples
→ Large number of negative ($y = 0$) examples		
→ Many different types of anomalies. Hard for any algorithm to learn from the positive examples what the anomalies look like		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.
→ Future anomalies may look nothing like any of the anomalous examples previously seen.		

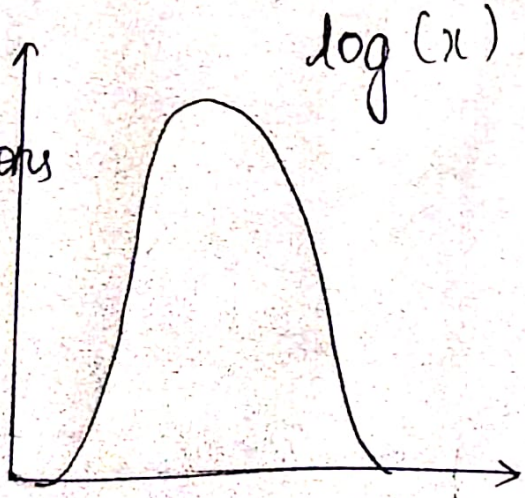
CHOOSING FEATURES



→ Gaussian distributed
$p(x; \mu, \sigma^2)$



transform using transformations so it looks more Gaussian



⑧ Error analysis for anomaly detection

We need

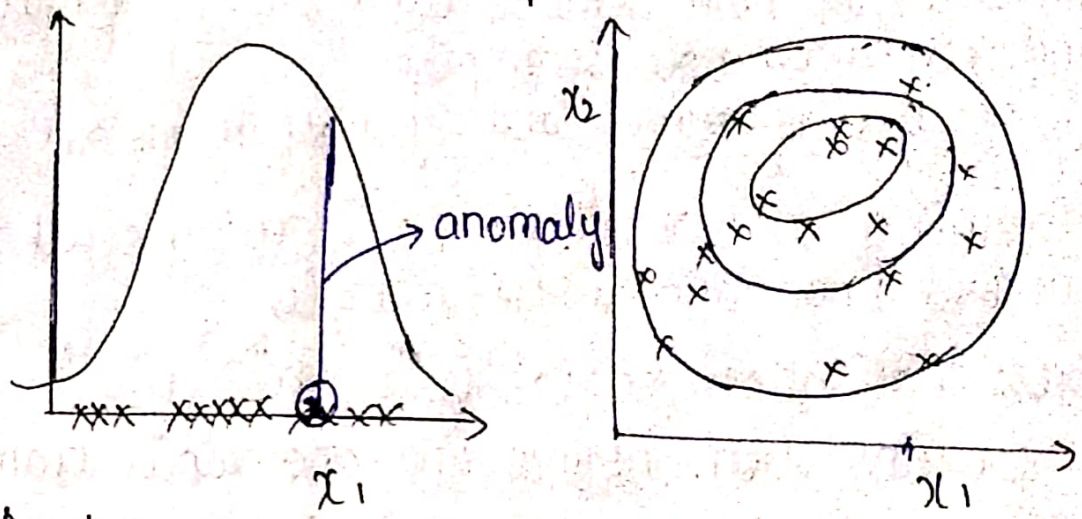
$p(x)$ is large for normal examples x.

$p(x)$ is small for anomalous examples x.

Most common problem :-

$p(x)$ is comparable for both normal and anomalous examples.

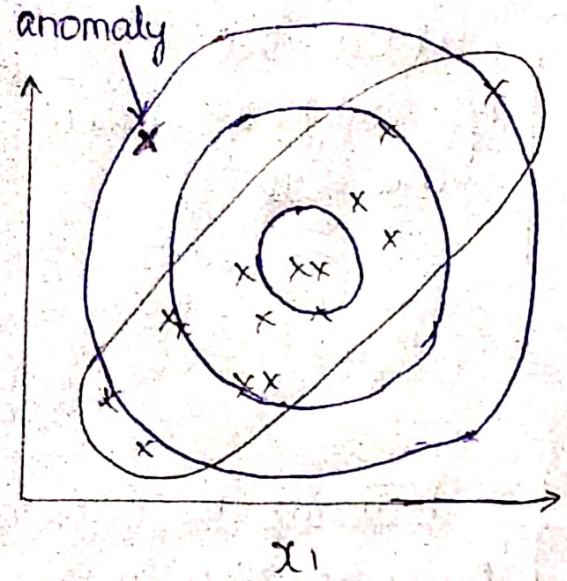
$x \rightarrow$ anomaly



∀ Analyse the failed cases and come up with features that distinguishes normal and anomaly. Choose features that might take on unusually large or small values in case of an anomaly.

MULTIVARIATE GAUSSIAN DISTRIBUTION

Problem :-



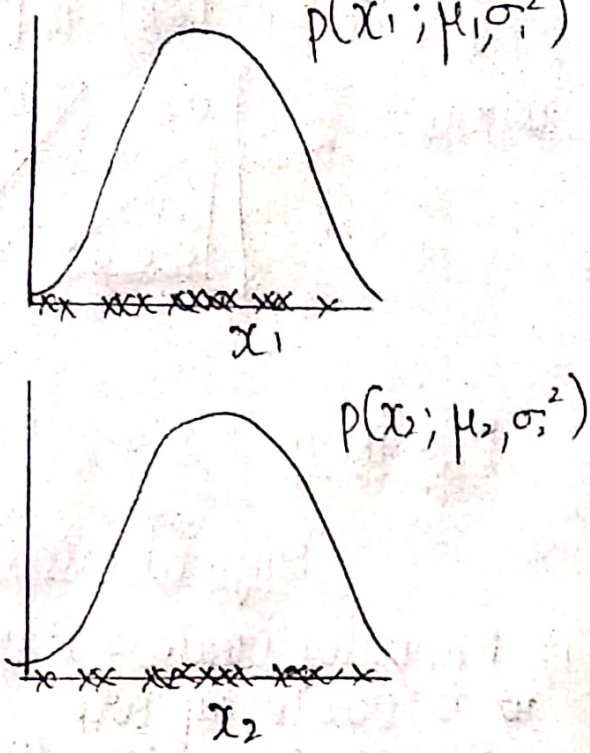


fig 1 :-

From fig 1, we can see that x_1 and x_2 vary linearly but due to univariate gaussian distribution, the test case x is falsely detected as a normality instead of an anomaly

---

→ Given $x \in R^n$, Don't model $p(x_1), p(x_2), \ldots$ etc seperately.

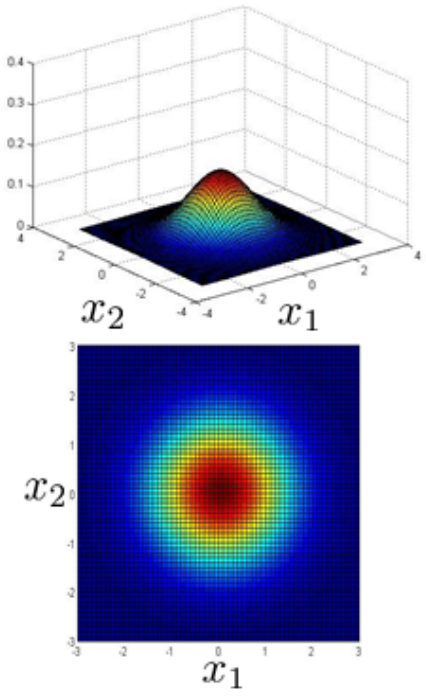
→ Model $p(x)$ all in one go.

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

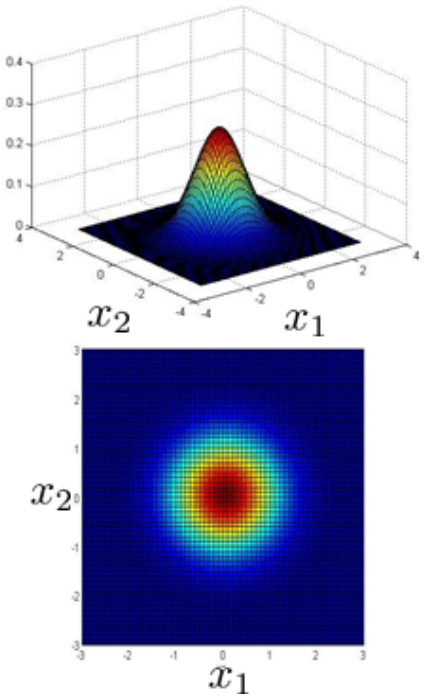
$$\therefore \boxed{p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)}$$

Multivariate Gaussian (Normal) examples

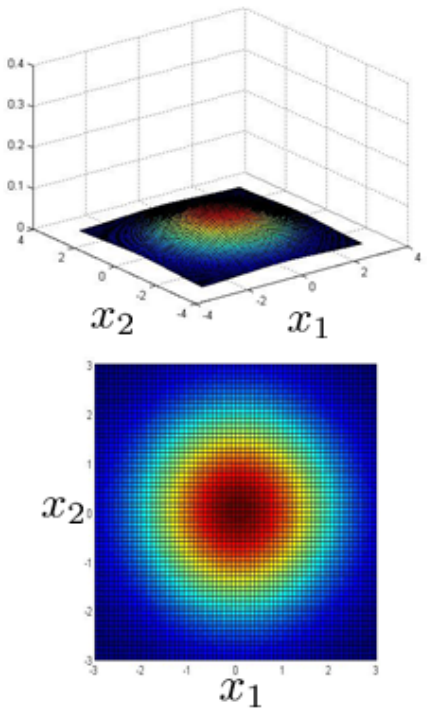
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$



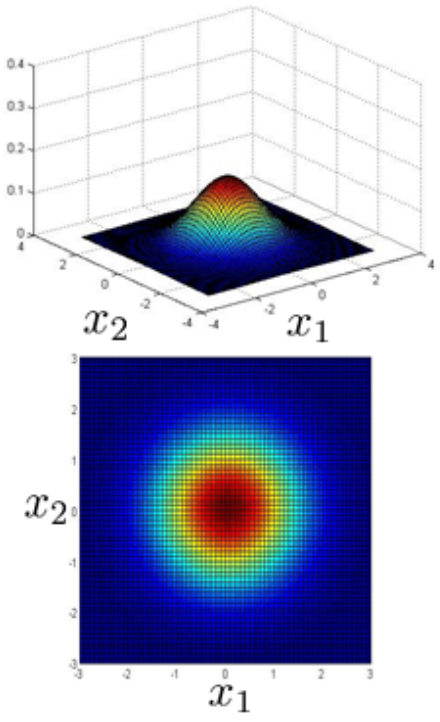
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix}$



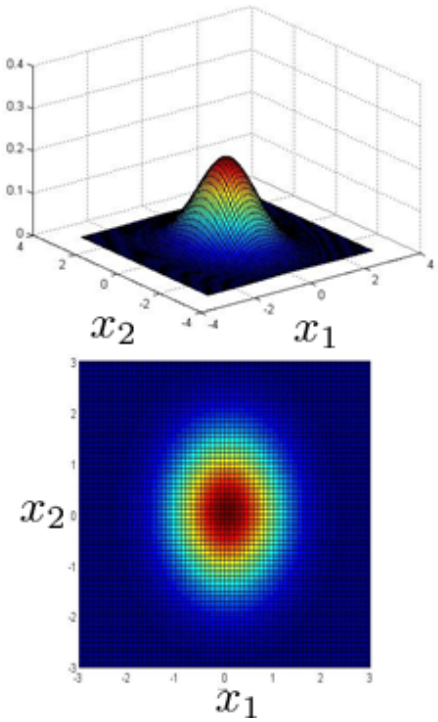
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$



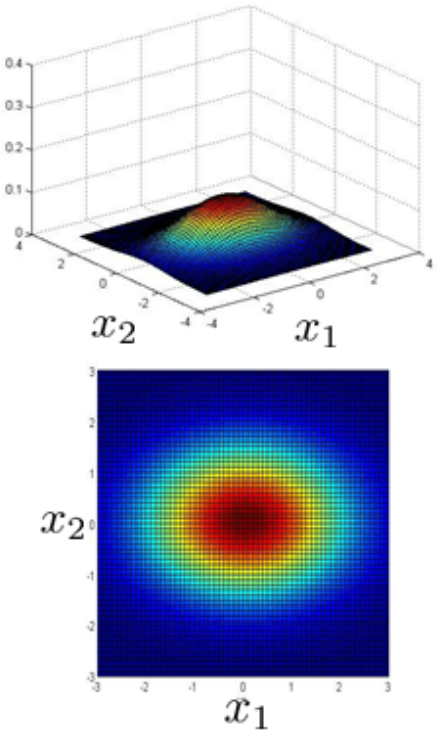
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$



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

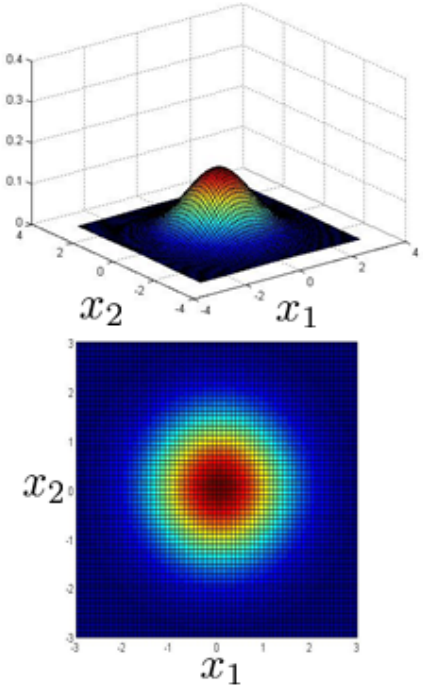


$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$

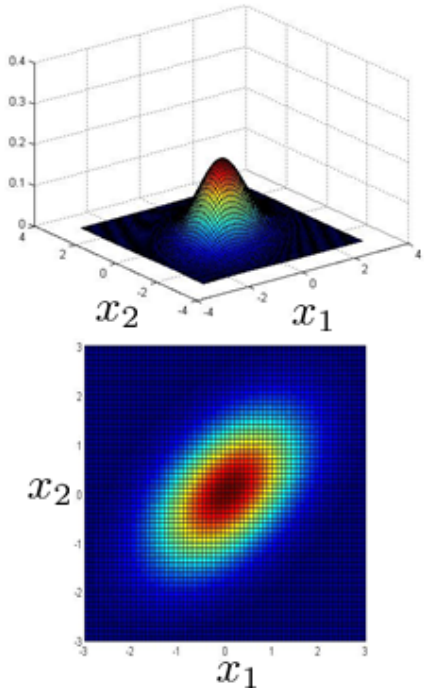


Multivariate Gaussian (Normal) examples

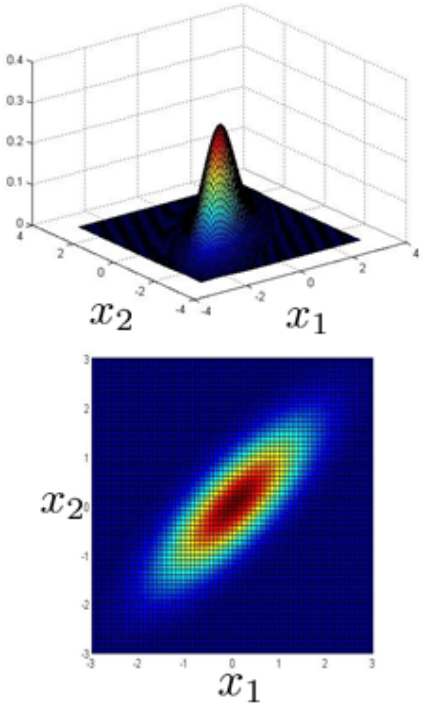
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$



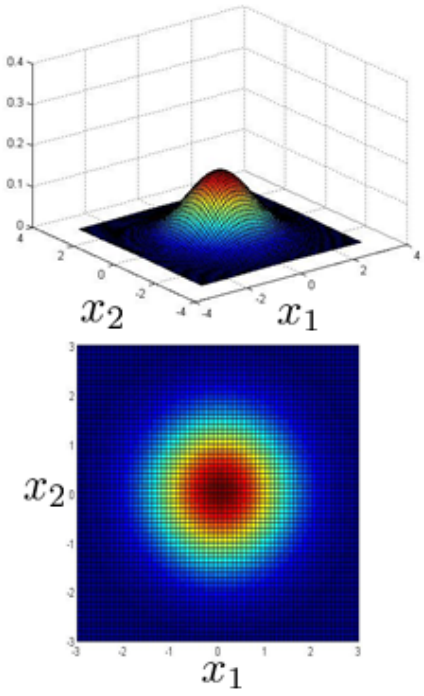
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$



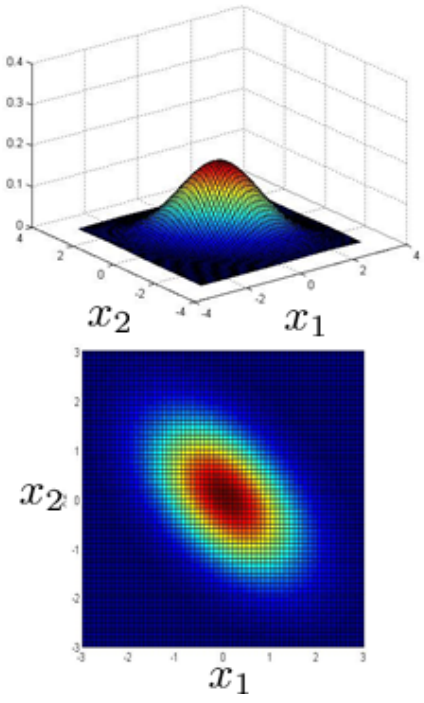
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$



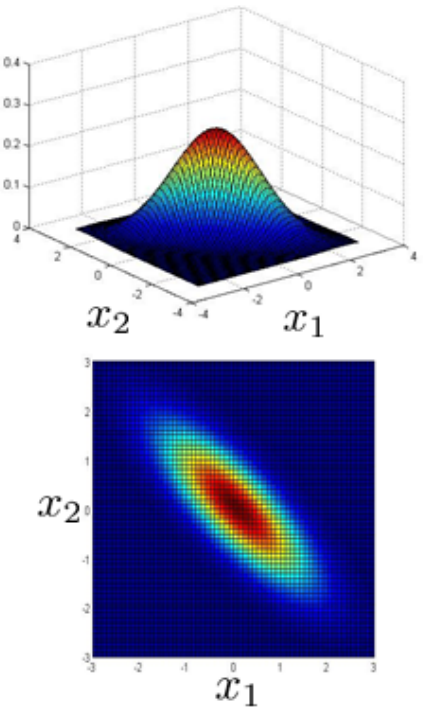
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$



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

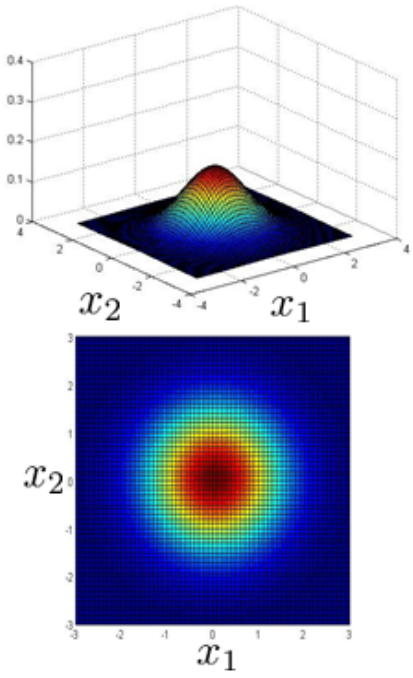


$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}$

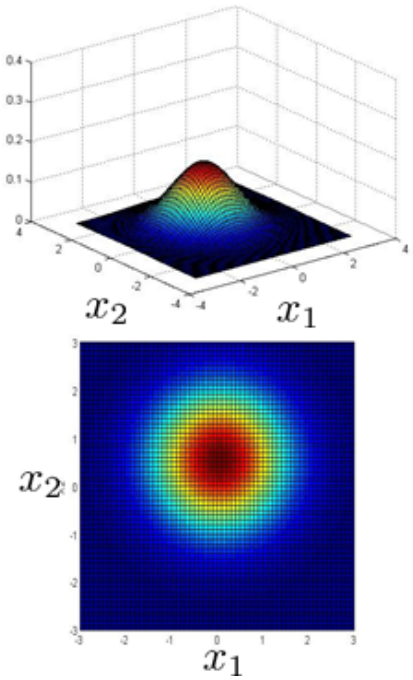


Multivariate Gaussian (Normal) examples

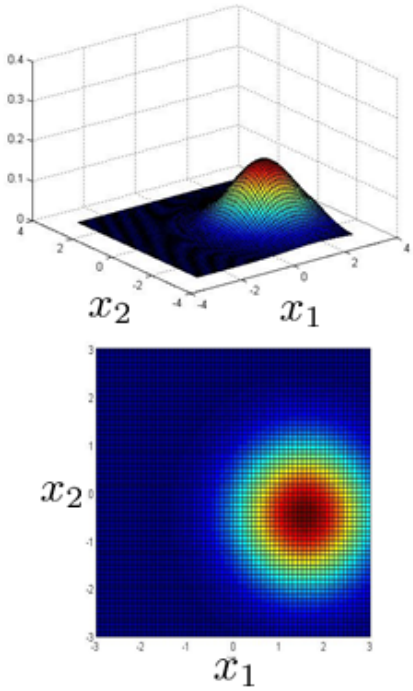
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$



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



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



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

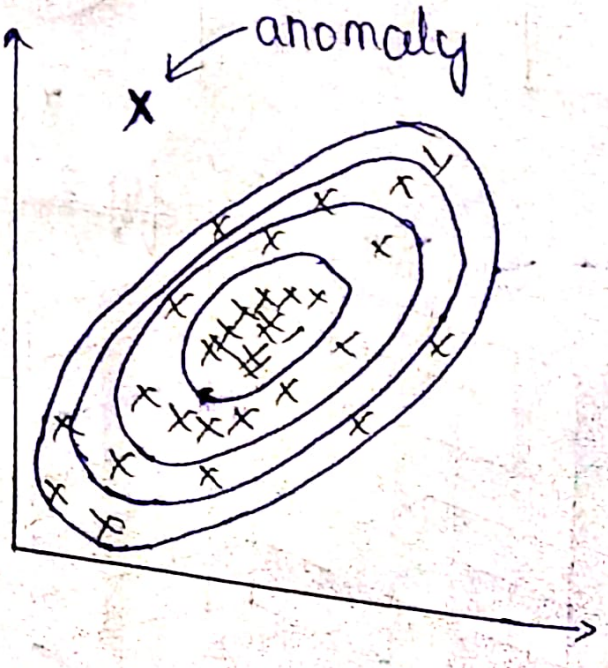
the parameters are

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

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

Anomaly detection with multivariate Gaussian

1) Fit the model $p(x)$ by setting μ and Σ.
2) Given new example x, compute its $p(x)$
3) Flag anomaly if $p(x) < \varepsilon$ else normal.



The original model corresponds to multivariate Gaussian

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

where $\Sigma = \begin{bmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots & \\ & & & \sigma_n^2 \end{bmatrix}$ where the off-diagonal elements are zero.

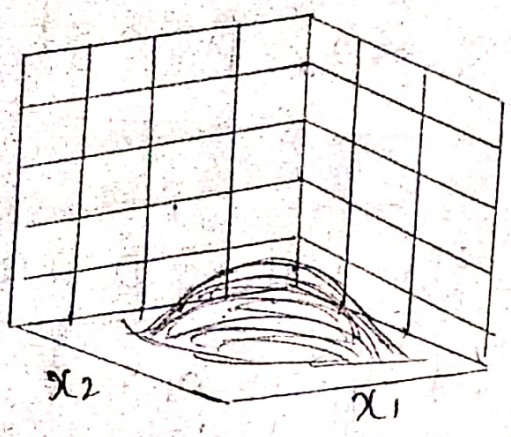
Difference

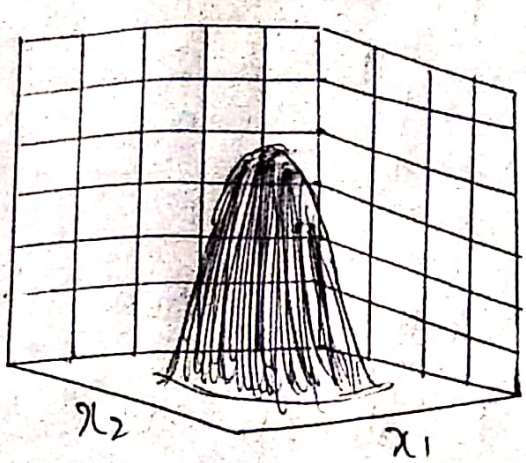
original model	Multivariate Gaussian
1) Manually create features to ~~create~~ capture anomalies where the features take unusual combinations.	Automatically captures correlations between features
2) Computationally cheaper	Computationally more expensive
3) Works even if training set size is small	Must have training set size greater than number of features.

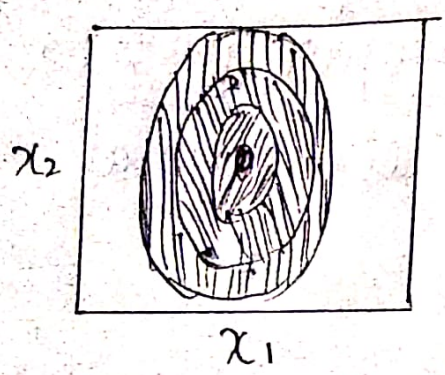
Relationship to original model

Original model :-

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







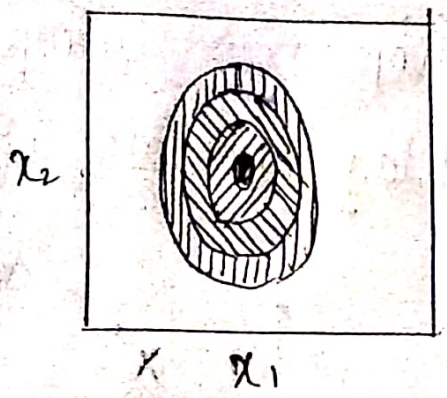
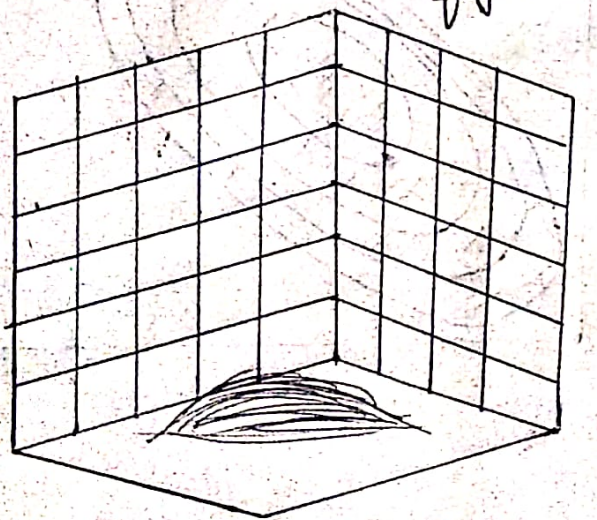
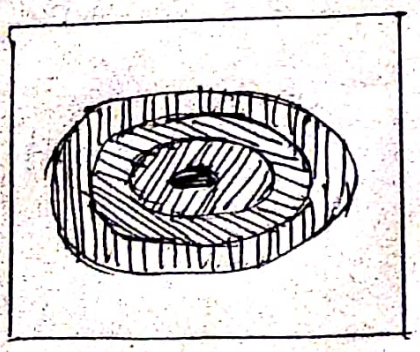


fig 1

fig 2





The original model corresponds to multivariate Gaussian

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

where $\Sigma = \begin{bmatrix} \sigma_1^2 & & & 0 \\ & \sigma_2^2 & & \\ & & \ddots & \\ 0 & & & \sigma_n^2 \end{bmatrix}$ where the off-diagonal elements are zero.

Difference

original model	Multivariate Gaussian
1) Manually create features to capture anomalies where the features take unusual combinations.	Automatically captures correlations between features
2) Computationally cheaper	Computationally more expensive
3) Works even if training set size is small	Must have training set size greater than number of features.