15. Anomaly Detection

Anomaly detection example:

Aircraft engine features:

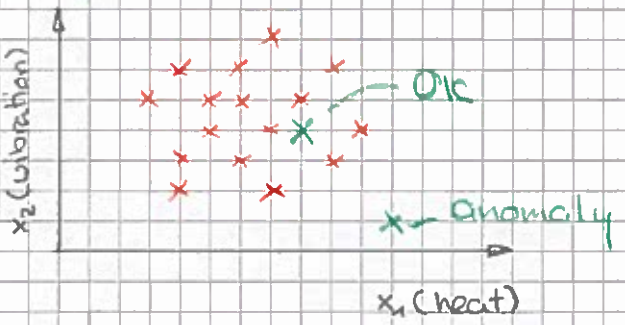
x_1 = heat generators

x_2 = vibration intensity

...

Data set: $x^{(1)}, x^{(2)}, \dots, x^{(m)}$

New Engine: x_{Test}

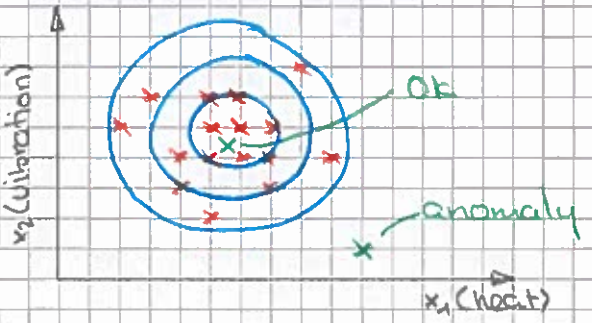


Density estimation:

Data set: $x^{(1)}, x^{(2)}, \dots, x^{(m)}$

Is x_{Test} anomalous?

$\Rightarrow$ Model $p(x)$



$p(x_{test}) < \varepsilon \Rightarrow$ flag anomaly

$p(x_{test}) \geq \varepsilon \Rightarrow$ flag ok

Anomaly detection example:

Fraud detection:

- $x^{(i)}$ = features of users activities
- Model $p(x)$ from data
- Identify unusual users by checking which have $p(x) < \varepsilon$

x_1 = typing speed
x_2 = unusual values
x_3 = ...
$p(x)$

Manufactoring

Monitoring computers in data center

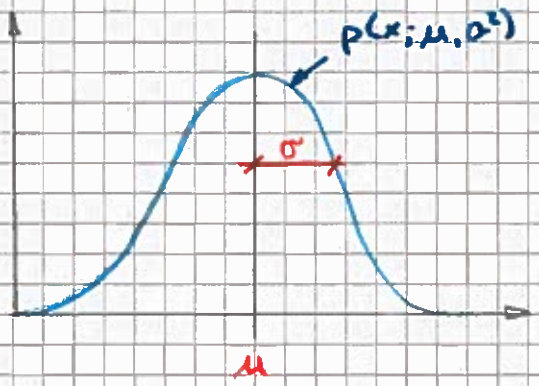
- $x^{(i)}$ = features of machine i
- x_1 = memory use, x_2 = # disk access/sec
  x_3 = CPU load, x_4 = CPU load / netw. traffic ...

$p(x) < \varepsilon$

Gaussian (Normal) distribution

Say $x \in \mathbb{R}$ if x is a distributed Gaussian with mean μ, variance σ^2

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

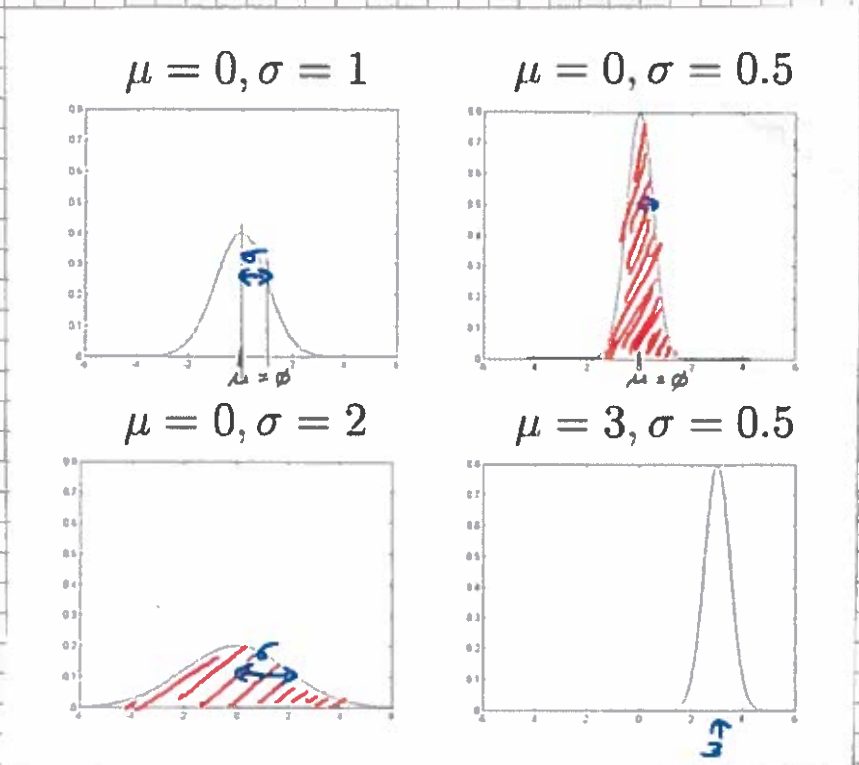


σ = Standard deviation

μ = mean

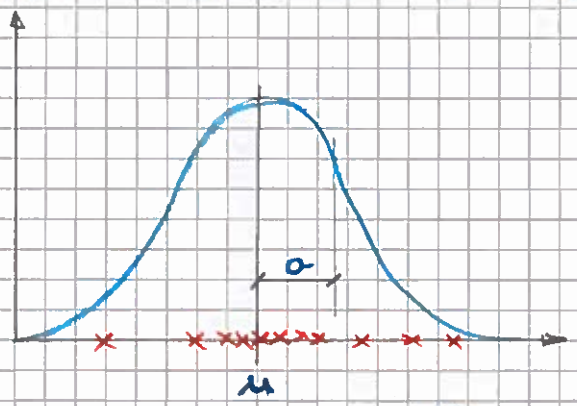
$$p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Gaussian distribution example



Parameter estimation

Data set: $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ $x^{(i)} \in \mathbb{R}$



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

mean $= 0$ $\mu = \frac{1}{m} \sum_{i=1}^{m} x^{(i)}$

variance $= 0$ $\sigma^2 = \frac{1}{m} \sum_{i=1}^{m} (x^{(i)} - \mu)^2$

average stand error

Density estimation:

Training set: $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$

Each example is $x \in \mathbb{R}^n$

$x_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$

$x_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$

$\vdots$

$p(x)$

$= p(x_1; \mu_1, \sigma_1^2) \cdot p(x_2; \mu_2, \sigma_2^2) \cdot p(x_3; \mu_3, \sigma_3^2) \dots p(x_n; \mu_n, \sigma_n^2)$

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

Algorithm:

1. Choose features x_i that you think might be indicative of anomalous examples. $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$

2. Fit parameters $\mu_1, \dots \mu_n \; \sigma_1^2, \dots \sigma_n^2$ $\Rightarrow p(x_j; \mu_j, \sigma_j^2)$

$$\mu_j = \frac{1}{m} \sum_{i=1}^{m} x_j^{(i)}$$

vectorized $= \mu \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix}$

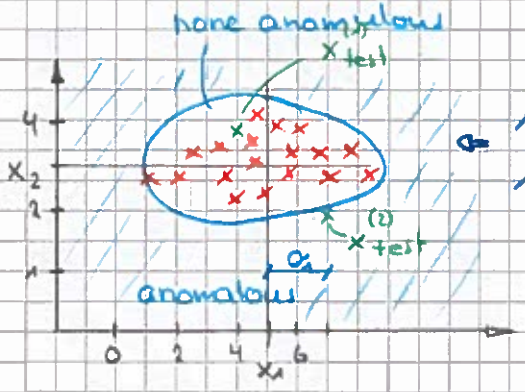
$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^{m} (x_j^{(i)} - \mu_j)^2$$

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

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

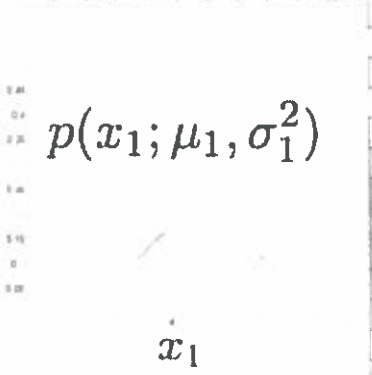
Anomaly if $p(x) < \varepsilon$

Anomaly detection example

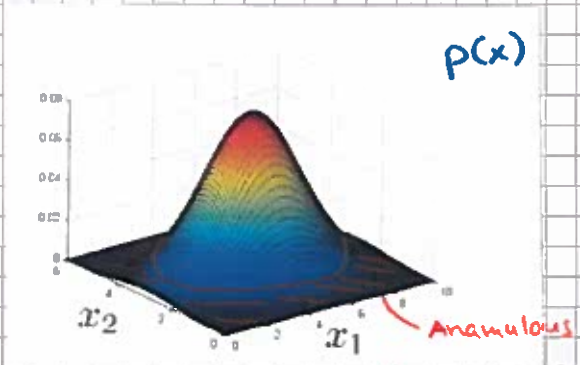


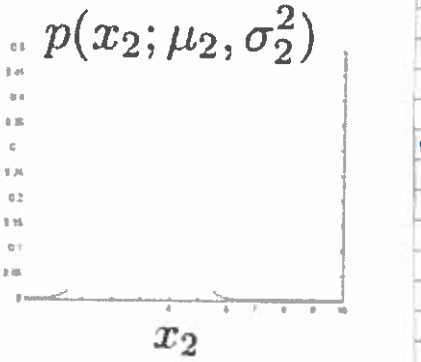
$\mu_1 = 5 ; \sigma_1 = 2$

$\mu_2 = 3 ; \sigma_2 = 1$



$p(x_1; \mu_1, \sigma_1^2$





$p(x_2; \mu_2, \sigma_2^2$

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

$\Rightarrow \varepsilon = 0,02$

$p(x_{test}^{(1)}) = 0,0426 \geq \varepsilon$

$p(x_{test}^{(2)}) = 0,0021 < \varepsilon$

The importance of real-number evaluation

When ~~learning~~ 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: $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ (assume normal examples / not anomalous)

Cross validation set: $(x_{cv}^{(1)}, y_{cv}^{(1)}) \dots, (x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})})$

Test set: $(x_{test}^{(1)}, y_{test}^{(1)}), \dots, (x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$

↳ Add some anomalous test set $y=1$

Aircraft engines motivating examples

10'000 good (normal) engines

20 flawed engines (anomalous) → 2-50 $y=1$

Training set: 6000 good engines ($y=0$) → $\mu_1, \sigma_1^2, \dots, \mu_n, \sigma_n^2$ $\quad p(x) = p(x_1; \mu_1, \sigma_1^2) \dots p(x_n; \mu_n, \sigma_n^2)$

CV: 2000 good engines ($y=0$); 10 anomalous ($y=1$)

Test: 2000 good engines ($y=0$); 10 anomalous ($y=1$)

Alternative, but not recommended:

Training set: 6000 good eng.

CV: 4000 good eng, 10 anomalous.

Test 4000 good eng, 10 anomalous

Algorithm evaluation:

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

On a cross validation / test set examples x, predict

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

$(x^{(i)}_{test}, y^{(i)}_{test})$

(data is very skewed $y=0 \gg y=1$ data)

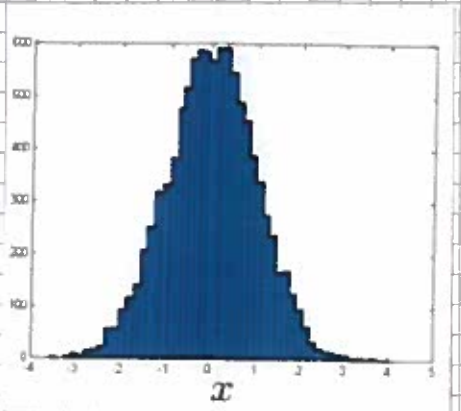
Possible evaluation metrics:

- True positiv, false positive, false negative, true negative
- precision Rate
- F_1-score

Can also use the cross validation set to use (choose) parameter ε

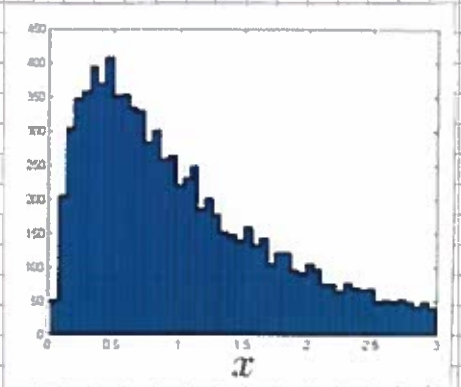
Anomaly Detection	Supervised learning
- Very small number of positive examples ($y=1$). 0-20 is common.	- Large number of positive & negative examples
- Large number of negative ($y=0$) examples	
- Many different "types" of anomalies. Hard for any algorithm to learn from positive examples what anomalies look like;	- Enough positive examples for algorithm to get a sense of what positive examples are like, future pos. examples likely to be similar to ones in training set.
- Future anomalies may look nothing like any of the anomalous examples we have seen so far.	
Application types:	
- Fraud detection	- eMail spam classification
- Manufacturing (e.g. Aircraft engines)	- Weather prediction (sunny/rainy/etc.)
- Monitoring machines in a data center...	- Cancer classification
	...

Non gaussian features



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

<= hist



do log(x)
=>

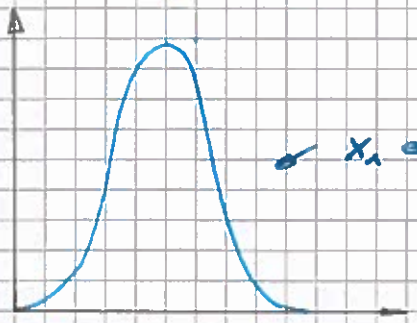
Possible functions examples:

$x_1 \leftarrow \log(x)$

$x_2 \leftarrow \log(x_2 + 1) = \log(x_2 + c)$

$x_3 \leftarrow \sqrt{x_3} = x_3^{\frac{1}{3}}$

$x_4 \leftarrow x_4^{\frac{1}{3}}$



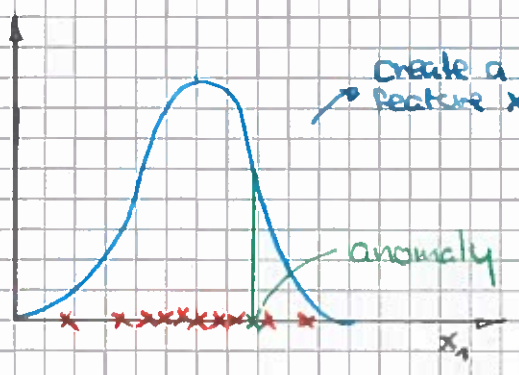
hist(x.^0.5, 50) => xNew = x.^0.05

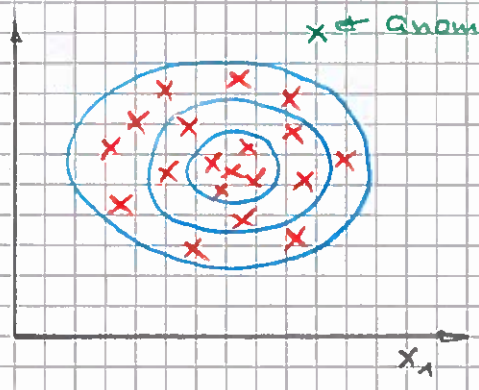
hist(log(x), 50) => xNewlog = log(x)

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.





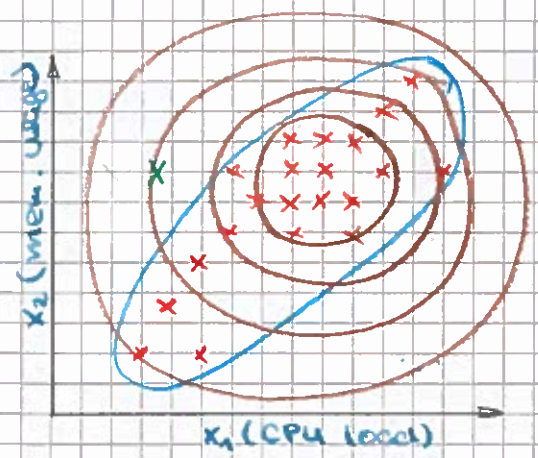
Should inspire you to create new features which indicate anomalies.

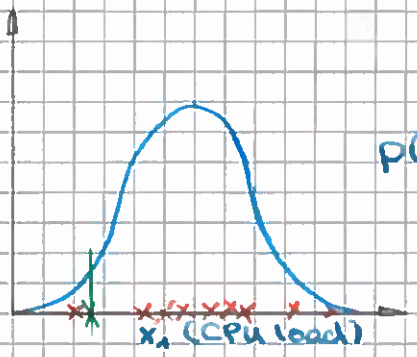
Monitoring computers in a data center:

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

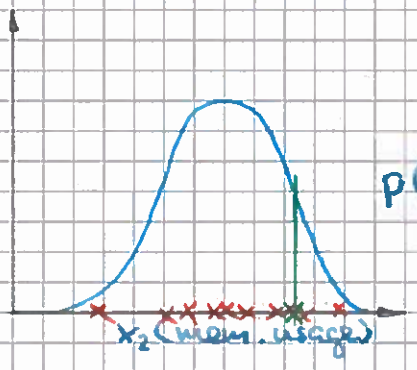
x_1 = memory use of computer
x_2 = # disk access / sec
x_3 = CPU load
x_4 = network traffic

create new feature $= 0$ $x_5 = \frac{\text{CPU load}}{\text{netw. traffic}}$; $x_6 = \frac{(\text{CPU load})^2}{\text{netw. traffic}}$





$p(x_1; \mu_1, \sigma_1^2)$



$p(x_2; \mu_2, \sigma_2^2)$

Anomaly detection algorithm will fail to flag x as anomaly

Multivariante Gaussian (Normal) distribution

$x \in \mathbb{R}^n$. Don't model $p(x_1), p(x_2), \ldots$, etc. separately

Model $p(x)$ all in one go

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

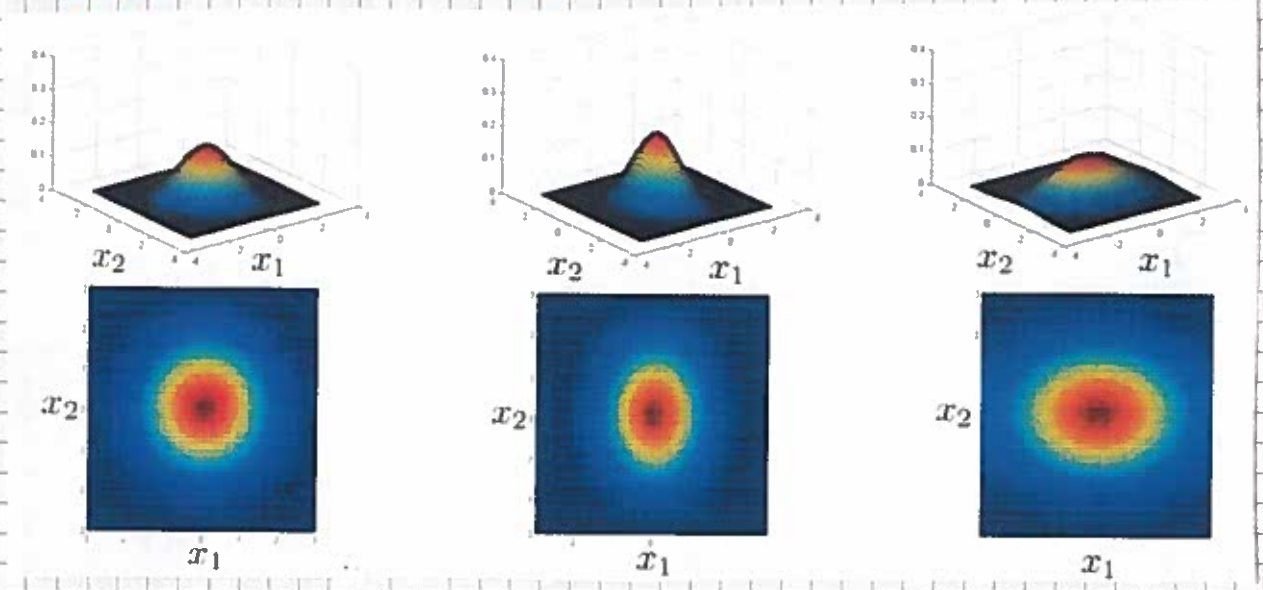
$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} \cdot |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

↳ $|\Sigma|$ = determinant of Σ = octave = $\det(\Sigma)$

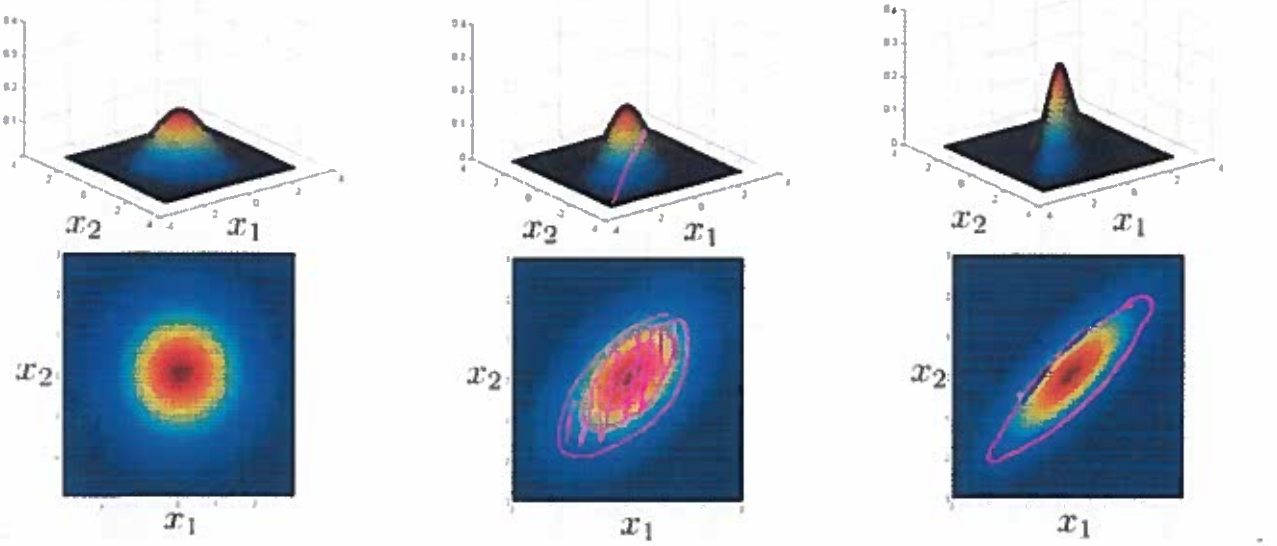
Gaussian Examples

Multivariate Gaussian Examples:

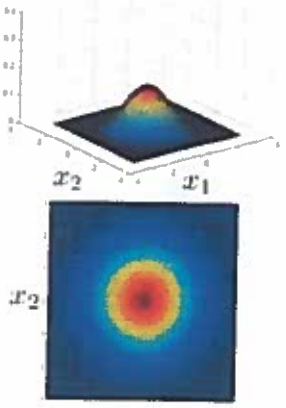
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $\quad \mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix} \Rightarrow \lambda$ $\quad \mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ ↑ λ



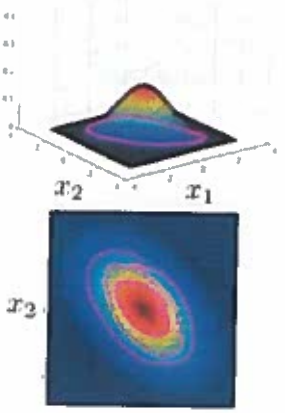
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $\quad \mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$ $\quad \mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1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



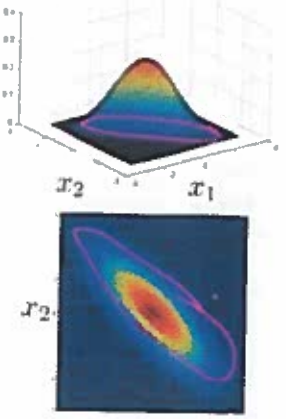
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$



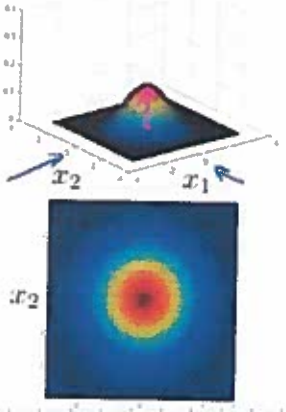
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$



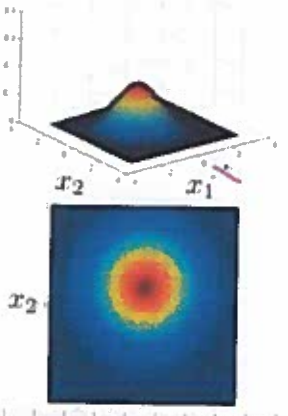
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}$



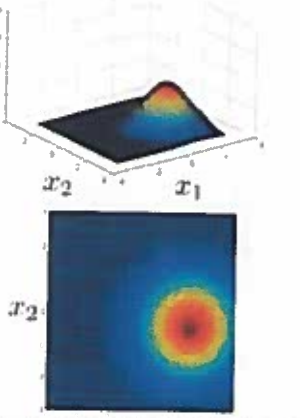
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$



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



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



Multivariate Gaussian (Normal) distribution

Parameters: $\mu, \Sigma \quad \mu \in \mathbb{R}^n \quad \Sigma \in \mathbb{R}^{n \times n}$

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

Parameter fitting:

Given training set: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\} \quad x \in \mathbb{R}^n$

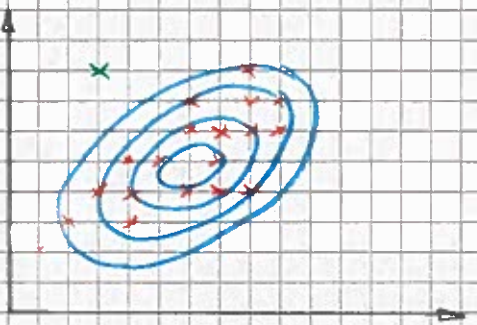
$$\mu = \frac{1}{m} \sum_{i=1}^{m} x^{(i)} \qquad \Sigma = \frac{1}{m} \sum_{i=1}^{m} (x^{(i)} - \mu)(x^{(i)} - \mu)^T$$

Anomaly detection with 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 anomaly if $p(x) < \varepsilon$

Relationship to original model:

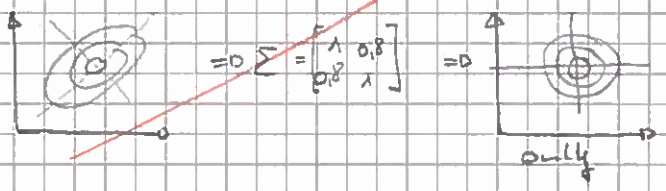
Original model: $p(x) = \prod_{i=1}^{n} (p(x_i; \mu_i, \sigma_i^2)$

Corresponds to multivariate Gaussian

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

where the contours are axes aligned (centered) x_1 / x_2.

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} \Rightarrow$$



Original Model	Multivariate Gaussian
- Manually create features to capture anomalies where x_1, x_2 take unusual combinations of values. $x_3 = \frac{x_1}{x_2} = \frac{\text{CPU load}}{\text{Netw. usage}}$	- Automatically captures correlations between features $\Sigma \in \mathbb{R}^{n \times n} \Rightarrow \Sigma^{-1}$
- Computationally cheaper (alternatively, scales better to large n) $n = >10'000, 100'000$	- Computationally more expensive $\Sigma \sim \frac{n^2}{2}$
- OK even if m (training set size) is small.	- Must have $m > n$, or else Σ is non-invertable (matrix singular) $m \geq 10 \cdot n$

$\Rightarrow$ $x_1 = x_2$, $x_3 = x_4 + x_5$ } Redundanten features; Σ is not invertible

1) $m > n$
2) get rid of redundanten feature