Week 9

Anomaly detection - problem motivation

Anomaly detection example:

Aircraft engine features:

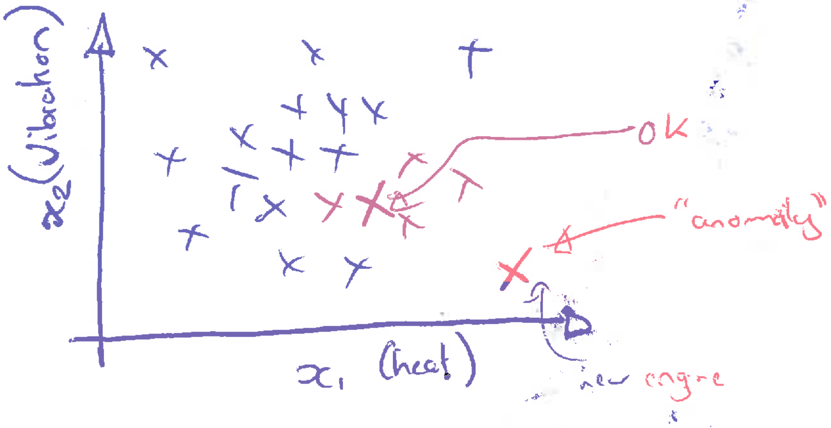
x_1 = heat generated

x_2 = vibration intensity

...

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

New engine: x_{test}



Density estimation

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

Is x_{test} anomalous? → build model $p(x)$

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

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

Week 9

Anomaly detection – problem motivation

Anomaly detection example:

Fraud detection:

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

model $p(x)$ from data.

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

Manufacturing

Monitoring computers in a data centre

$x^{(i)}$ = features of machine i

x_1 = memory use, x_2 = number of disk accesses / sec,

x_3 = CPU Load, x_4 = CPU Load / network traffic.

$p(x) < \varepsilon$

Week 9

Anomaly detection - Gaussian distribution

→ also called "normal distribution"

Gaussian (Normal) distribution

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

← bell shape curve parameterized by:

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

↑ "distributed as"

sigma or "one standard deviation"

σ

parameterizes x taking on different values.

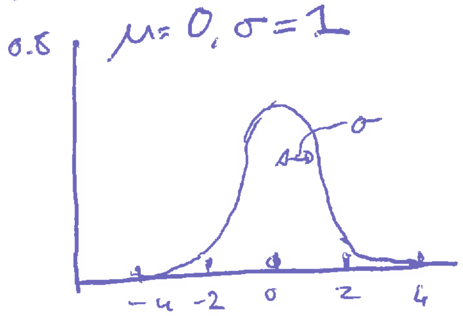
$$p(x; \mu, \sigma^2)$$

probability of x is is parameterized by the two parameters μ e σ^2

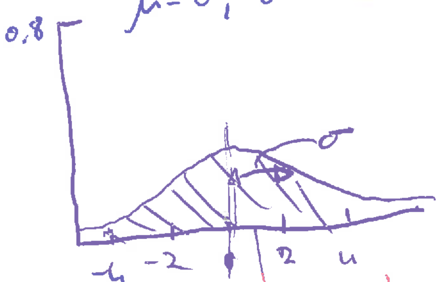
$$= \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Gaussian distribution example

$\mu = 0, \sigma = 1$



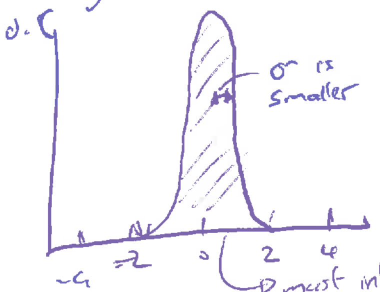
$\mu = 0, \sigma = 2$.



must integrate 1

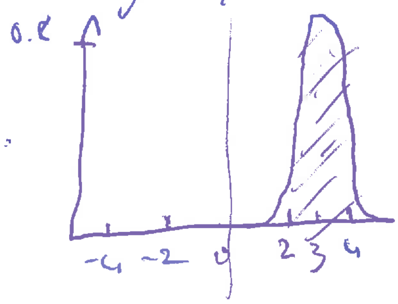
$\mu = 0, \sigma = 0.5$

$\sigma^2 = 0.25$



must integrate to 1

$\mu = 3, \sigma = 0.5$



Week 9

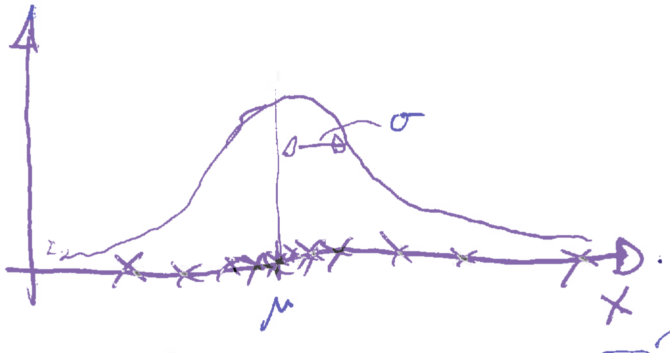
Parameter estimation

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

distributed

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

normal distribution ? ?



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

ML

m−1

both are the "maximum likely estimations"

Week 9

Anomaly ~~de~~ detection – Algorithm

Density estimation

Training set $\{x^{(1)}, \ldots, x^{(m)}\}$

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

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

$x_2 \sim N(\mu_2, \sigma_2)$

$x_3 \sim N(\mu_3, \sigma_3)$

$p(x)$

$$= \boxed{p(x_1;\mu_1,\sigma_1^2)\,p(x_2;\mu_2,\sigma_2^2)\,p(x_3 \quad)\ldots p(x_n \quad)}$$

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

← compact form

product of a set of values

similar to Σ except product....

i.e. $\sum_{i=1}^{n} i = 1+2+3+4 \ldots n$

$\prod_{i=1}^{n} i = 1 \times 2 \times 3 \times \ldots \times n.$

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

Anomaly detection algorithm

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

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

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

parameterized by $p(x_j;\mu_j,\sigma_j^2)$

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

estimate all the values for μ simultaneously

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix} = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$$

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

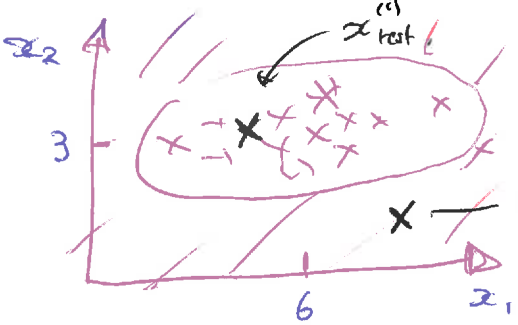
$$p(x) = \prod_{j=1}^{n} p(x_j;\mu_j,\sigma_j^2) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}\,\sigma_j} \exp\left(-\frac{(x_j-\mu_j)^2}{2\sigma_j^2}\right)$$

Anomaly if $p(x) < \varepsilon$

Week 9

Anomaly detection – Algorithm

Anomaly detection example:

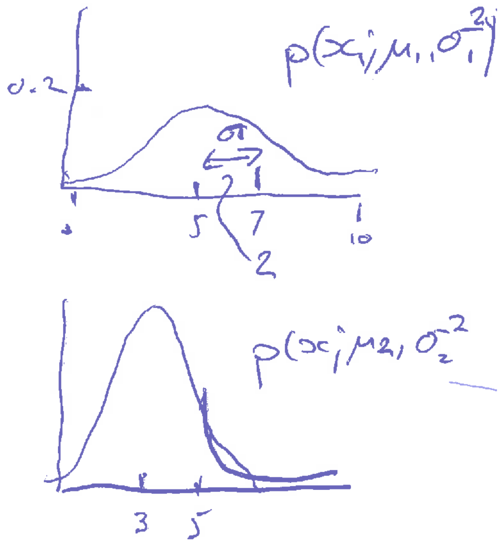


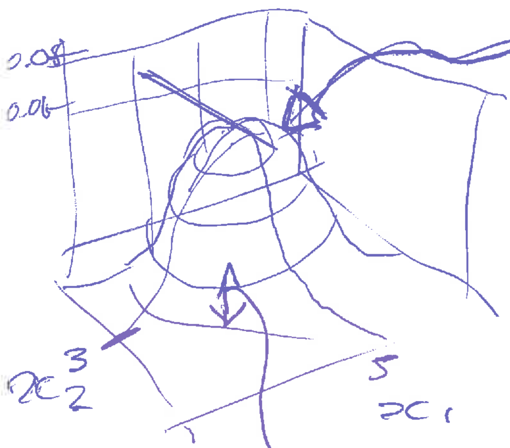
Is this an anomaly or not?

$\mu_1 = 5$, $\sigma_1 = 2$

$\mu_2 = 3$, $\sigma_2 = 1$

$\varepsilon = 0.02$





$p(x)$ which can be given by:

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

$\varepsilon = 0.02$

$p(x_{test}^{(1)}) = 0.0426 \geq \varepsilon$ $\therefore$ not an anomaly

$p(x_{test}^{(2)}) = 0.0021 < \varepsilon$ $\Rightarrow$ anomaly

all the points with a lot of height correspond to big "non-anamolous" areas.

Week 9

Anomaly detection - developing and evaluating an anomaly detection system

The importance of real-number evaluation

When 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 labelled data, of anomalous and non-anomalous examples ($y=0$ if normal, $y=1$ if anomalous)

Training set: $\underline{x^{(1)}, x^{(2)}, \dots x^{(m)}}$ (assume normal examples / not anomalous)

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

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

have a few examples where $y=1$ (anomalous)

Aircraft engines motivating example

10000 good (normal) engines

$\boxed{20}$ flawed engines (anomalous) $y=1$ (20 ... 50) ← typical anomalous example number

Training set: 6000 good engines ($y=0$)

CV: 2000 good engines ($y=0$), $\boxed{10}$ anomalous ($y=1$)

Test: 2000 good engines ($y=0$), $\boxed{10}$ " ($y=1$).

Use these to fit $p(x) = p(x_1; \mu_1, \sigma_1^2) \times p(x_2 \dots$

Alternative:

Training set: 6000 good engines

CV: 4000 good engines ($y=0$), 10 anomalous ($y=1$)

Test: 4000 " engines ($y=0$), 10 anomalous ($y=1$)

→ same data → not recommended.

Week 9

Anomaly detection – developing...

Algorithm evaluation

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

on a cross validation / test example x, predict

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

on the cross validation set we are going to use our model to predict y.

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

Possible evaluation metrics:

- True positive, false positive, false negative, true negative
- precision / recall
- F_1-score.

→ continually evaluate on the cross-validation set

↓

final evaluation on the test set.

Can also use cross-validation set to choose parameter ε

Week 9

Anomaly detection: anomaly detection vs. supervised learning algorithm.

Anomaly detection vs. **Supervised learning**

Anomaly detection

Very small number of positive examples ($y=1$). (0-20 is common)
Large number of negative ($y=0$) examples $p(x)$.

many different "types" of anomalies. Hard for any algorithm to learn from positive examples what the anomalies look like; future anomalies may look like any of the anomalies we have seen so far

(alot of $y=1$)

- fraud detection
- manufacturing (air craft engines)
- monitoring machines in a data centre

Supervised learning

Large number of positive and negative examples.

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.

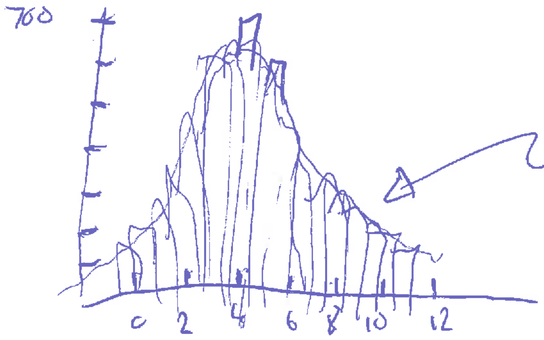
spam

- email spam classification
- weather prediction
- cancer classification.

Week 9

Anomaly detection - choosing what features to use

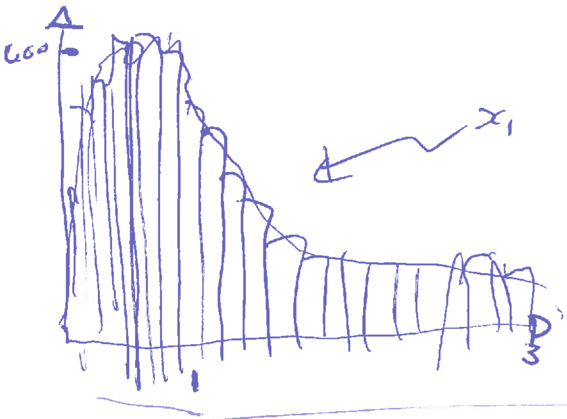
Non-gaussian features



$p(x_i; \mu_i, \sigma^2)$

hist command to plot a histogram in octave

hist(x, 50) 50 bins.

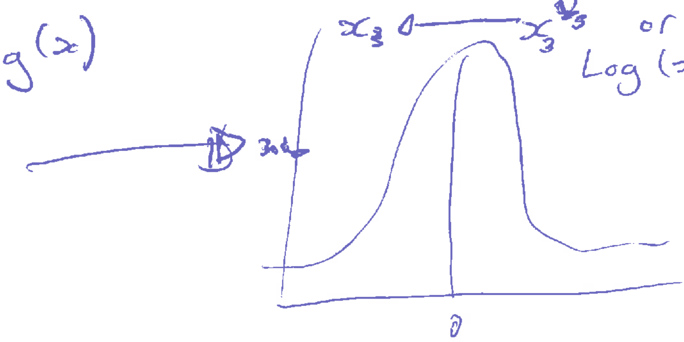


$\log(x)$

$x_1 \leftarrow \log(x_1)$

$x_2 \leftarrow \log(x_2 + 1)$ play with

$x_3 \leftarrow x_3^{1/3}$ or $\log(x_2 + C)$



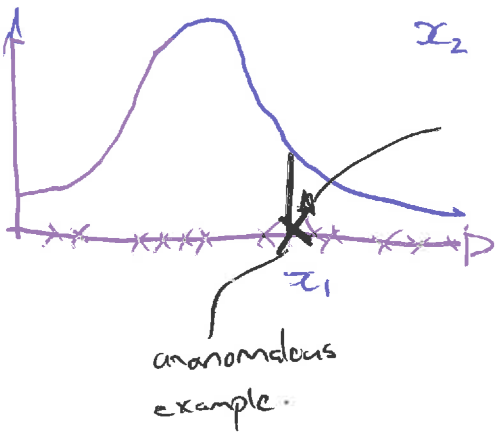
Error Analysis for anomaly detection

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

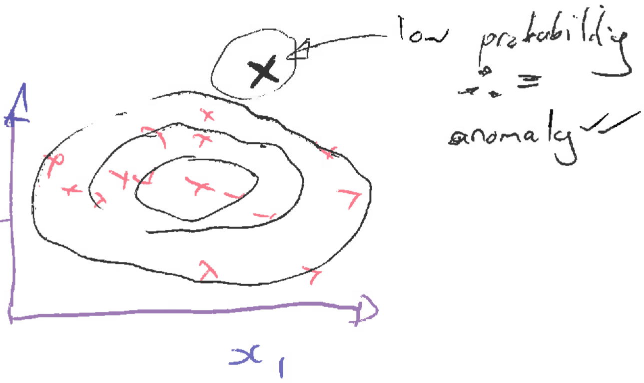
$p(x)$ small for anomalous examples x.

Most common problem:

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



remapping old examples onto new feature "x_2"



Week 9

Anomaly detection – choosing what features to use

Monitoring computers in a data centre

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

x_1 = memory use of computer

x_2 = number of disk accesses/sec

x_3 = CPU load

x_4 = network traffic

$$x_5 = \frac{\text{CPU LOAD}}{\text{Network traffic}}$$

in the case of an infinite loop CPU LOAD → ~~up~~
whilst network traffic → 0

↳ ∴ seems like a likely failure case

$$x_6 = \frac{(\text{CPU load})^2}{\text{network traffic}}$$

Week 9

Anomaly detection - Anomaly detection using the multivariate gaussian distribution

Multivariate Gaussian (normal) distribution

Parameter fitting:

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

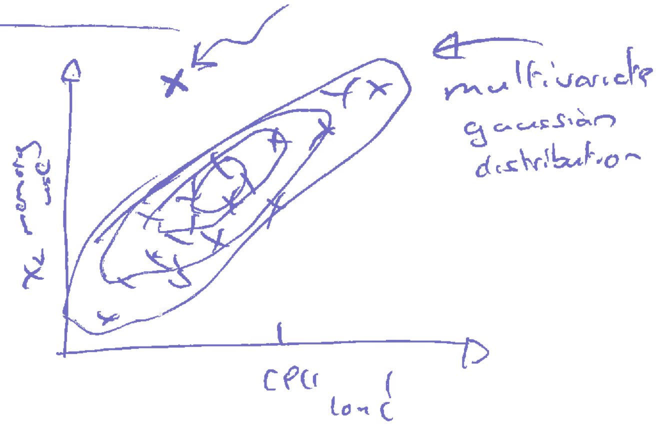
→ $\boxed{\mu} = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$ → $\boxed{\Sigma} = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)}-\mu)(x^{(i)}-\mu)^T$

Anomaly detection with the multivariate Gaussian

1) Fit model $p(x)$ by setting

→ $\mu =$

→ Σ



Flag an anomaly if $p(x) < \epsilon$

Relationship to original model

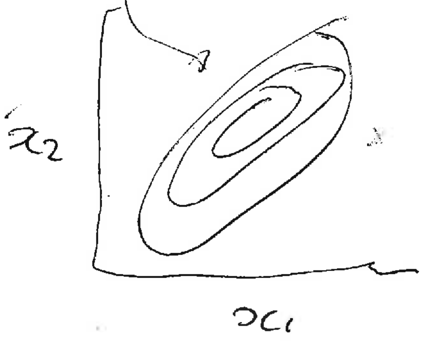
corresponds to multivariate Gaussian;

where $\Sigma = \begin{bmatrix} \sigma_1^2 & & 0 & \\ & \sigma_2^2 & & \\ 0 & & \sigma_3^2 & \\ & & & \ddots \end{bmatrix}$ ← exactly the same as the Gaussian model, however.

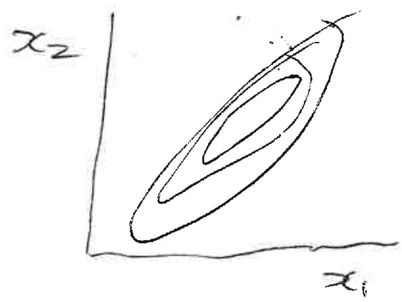
Original Model	Multivariate Gaussian
$p(x_1;\mu_1,\sigma_1^2) \times \cdots \times p(x_n;\mu_n,\sigma_n^2)$	$\frac{1}{(2\pi)^{\frac{n}{2}}\lvert\Sigma\rvert^{\frac{1}{2}}}\exp(-\ldots\Sigma^{-1}\ldots)$ → $\Sigma \to \mathbb{R}^{n\times n}$ ← computationally very expensive
• Manually create features to capture anomalies where x_1, x_2 take unusual combinations of values $x_3 = \frac{x_1}{x_2} = \frac{\text{CPU Load}}{\text{memory}}$	• Automatically captures correlations between features
Computationally cheaper (alternatively scales better to large n) $n = 100\,000$	• Computationally more expensive.
OK even if m (training set size) is small	Must have $\boxed{m} > n$, or else Σ is non-invertible. → look for redundant features → $\boxed{m \geq 10n}$ rule of thumb.

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

the ~~stat~~ covariance matrix skews the relationships between the variables

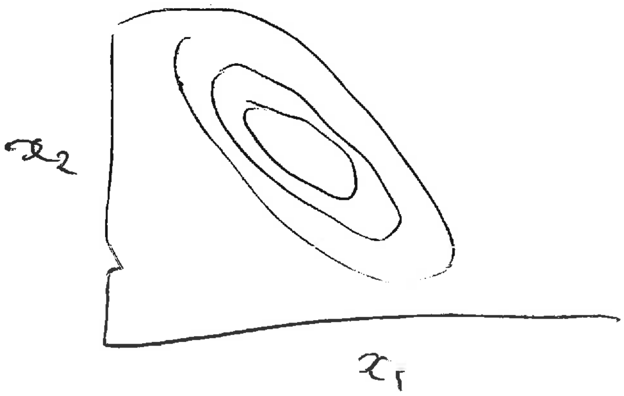


$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$



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

-ve correlation

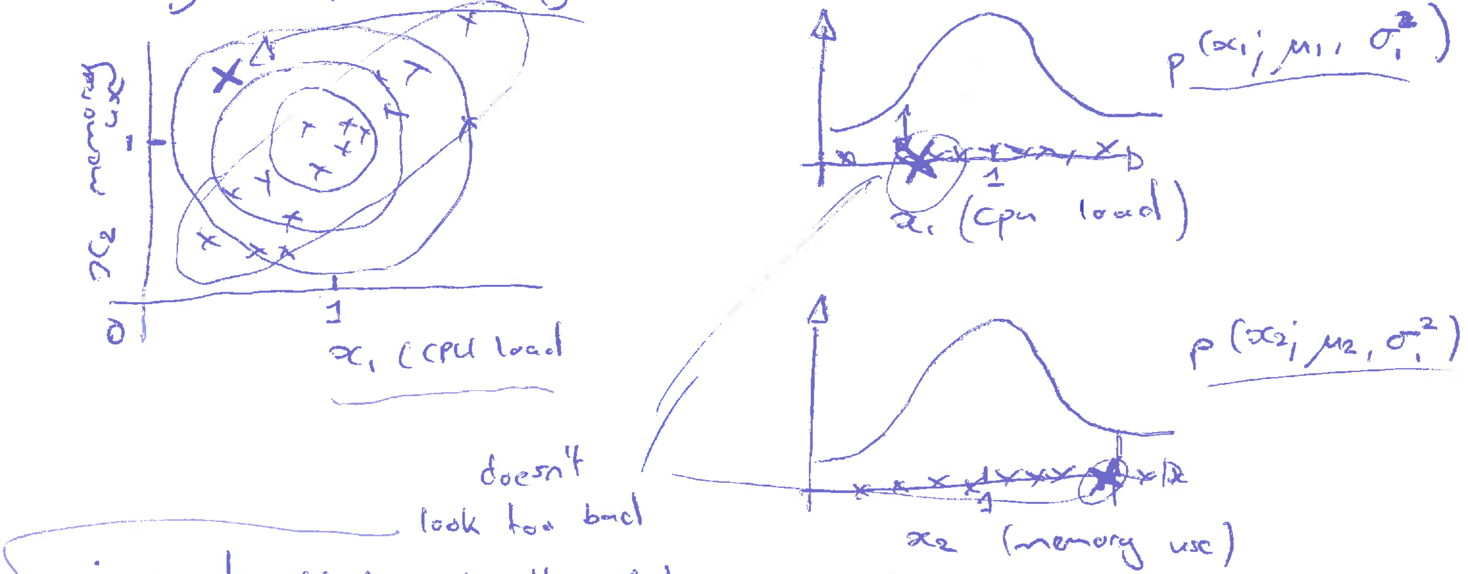


→ can also vary μ such that we move the peak or centre (shift it).

Week 9

Anomaly Detection - Multivariate Gaussian distribution

motivating example: monitoring machines in a data center



$\Rightarrow \therefore$ anomaly detection algorithm fails to classify data point.

Multivariate Gaussian (normal) distribution

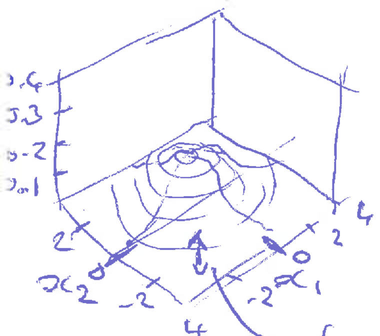
$x \in \mathbb{R}^n$, Don't model $p(x_1), p(x_2), \dots,$ etc separately.
Model $p(x)$ all in one go.
Parameters : $\mu \in \mathbb{R}^n$, $\Sigma \in \mathbb{R}^{n \times n}$ (covariance matrix)

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

$|\Sigma|$ ← determinant of Σ

Multivariate Gaussian (Normal) examples

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



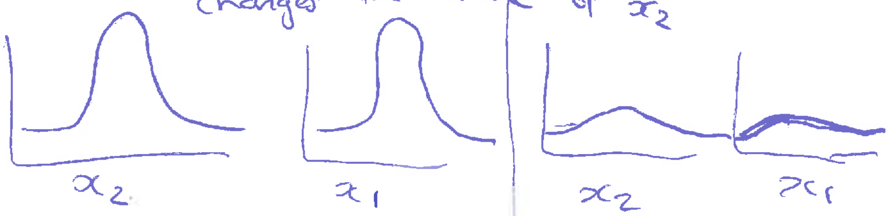
for a specific value of μ & Σ the height is $p(x)$.

changes the variance of x_1

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

changes the variance of x_2

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



Week 9

Recommender Systems: problem formulation

example: predicting movie ratings
user ratings zero to five stars.

one star ✱ ⟶ ✱✱✱✱✱ five stars.

Movie	Alice(1)	bob(2)	Carol(3)	Dave(4)
Love at last	5	5	0	0
Romance forever	5	? 4.5	? 0	0
Cute puppies of love	? 5	4	0	? 0
Nonstop car chases	0	0	5	4
Swords vs karate	0	0	5	? 4

$n_u = 4$ $n_m = 5$

n_u = no. of users
n_m = no. movies
$r(i,j) = 1$ if user j has rated movie i
$y^{(i,j)}$ = rating given by user j to movie i
(defined only if $r(i,j) = 1$.

Week 9

Recommender systems: Content based recommendations

Content based recommendation systems

Movie	$\theta^{(1)}$ Alice (1)	θ^{2} Bob (2)	θ^{3} Carol (3)	θ^{4} Dave (4)	x_1 (romance)	x_2 action
$x^{(1)}$ Love at last 1	5	5	0	0	0.9	0
Romance forever 2	5	?	?	0	1.0	0.01
$x^{(3)}$ Cute puppies of love	? 4.95	4	0	?	0.99	0
Nonstop car chases 4	0	0	5	4	0.1	1.0
Swords vs karate 5	0	0	5	?	0	0.9

$x^{(5)}$ ← feature vector.

$\rightarrow x^{(1)} = \begin{bmatrix} 1 \\ 0.9 \\ 0 \end{bmatrix}$

$\boxed{n = 2}$ ← number of features (degree of romance, degree of action)

For each user j, learn a parameter $\theta^{(j)} \in \mathbb{R}^3$. Predict user j as rating movie i with $(\theta^{(j)})^T x^{(i)}$ stars $\hookrightarrow \theta^{(j)} \in \mathbb{R}^{n+1}$

$$x^{(3)} = \begin{bmatrix} 1 \\ 0.99 \\ 0 \end{bmatrix} \begin{matrix} \leftarrow x_0 \\ x_1 \\ \leftarrow x_2 \end{matrix} \qquad \theta^{(1)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix} \qquad (\theta^{(1)})^T x^{(3)} = 5 \times 0.99 = 4.95$$

?

Problem formulation

$r(i,j) = 1$ if user j has rated movie i (0 otherwise)

$y^{(i,j)}$ = rating by user j on movie i (if defined)

θ^{j} = parameter vector for user j

$x^{(i)}$ = feature vector for movie i

for user j, movie i, predicting rating: $(\theta^{(j)})^T (x^{(i)})$

m^{j} = no. of movies rated by user j.

To learn $\theta^{(j)}$:

$$\min_{\theta^{(j)}} \frac{1}{2\cancel{m^{(j)}}} \underbrace{\sum_{i:r(i,j)=1}}_{} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2\cancel{m^{(j)}}} \sum_{k=1}^{n} (\theta_k^{(j)})^2$$

← summing over all the values that user j has rated.

← don't regularize over the bias term.

Week 9

Recommender System: Content based recommendations

To learn $\theta^{(j)}$ (parameter for user j)

$$\min_{\theta^{(j)}} \frac{1}{2} \sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{k=1}^{n} \left(\theta_k^{(j)} \right)^2$$

To learn $\theta^{(1)}, \theta^{(2)} \ldots \theta^{(n_u)}$ ← for all users

$$\min_{\theta^{(1)} \ldots \theta^{(n_u)}} \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} \left(\theta_k^{(j)} \right)^2$$

↑

∴ extra summation.

↳ in a nutshell gradient descent for recommendation...