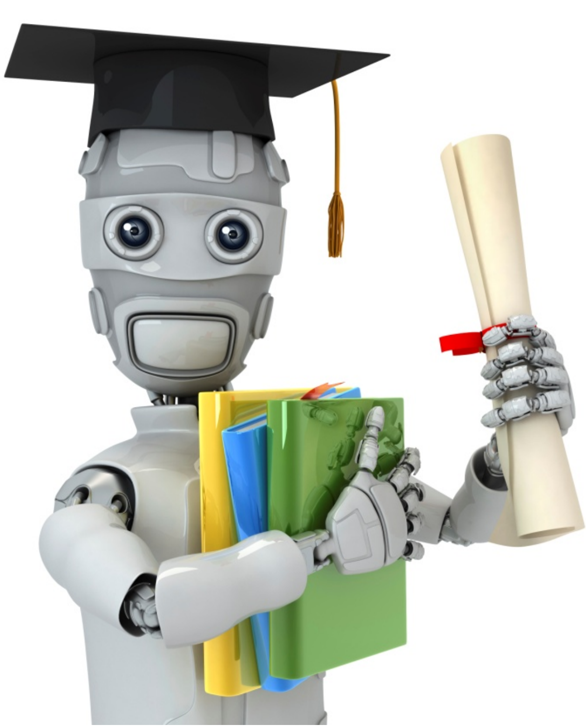


Machine Learning

Anomaly detection

Problem motivation

Anomaly detection (or outlier detection) is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data.

Unsupervised anomaly detection techniques detect anomalies in an unlabeled test data set under the assumption that the majority of the instances in the data set are normal by looking for instances that seem to fit least to the remainder of the data set.

Anomaly detection example

Aircraft engine features:

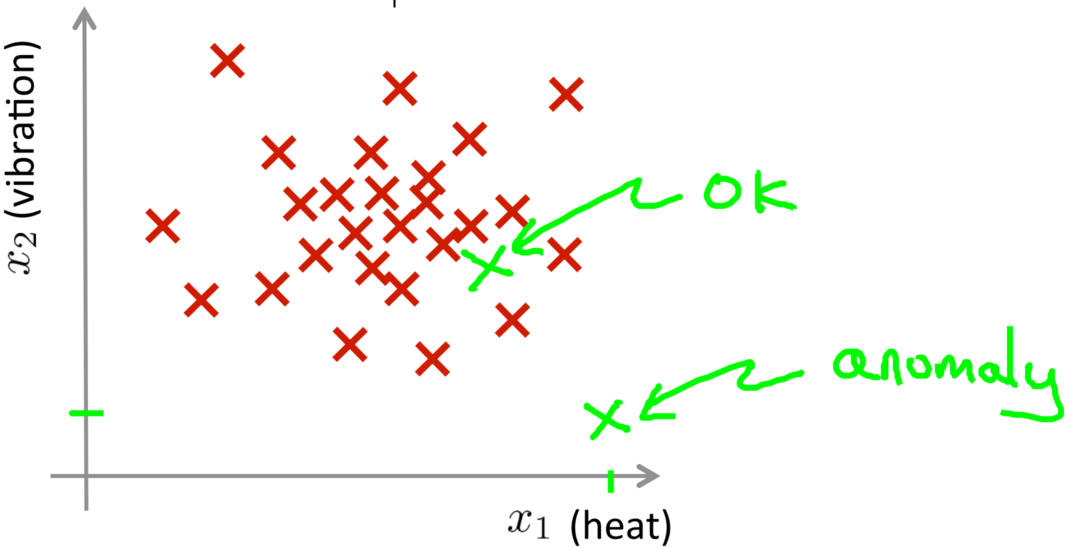
x_1 = heat generated

x_2 = vibration intensity

…

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

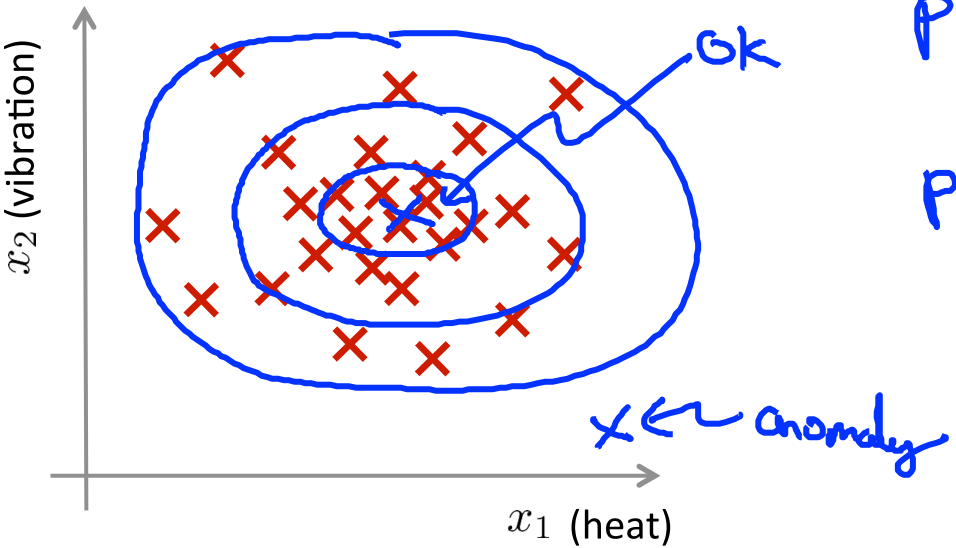
New engine: x_{test}



Density estimation

- Dataset: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$
- Is x_{test} anomalous?

Model $\underline{p(x)}$.



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

$p(x_{test}) \geq \varepsilon \rightarrow$ OK

Andrew Ng

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 $p(x) < \varepsilon$

x_1
x_2
x_3
x_4

$p(x)$

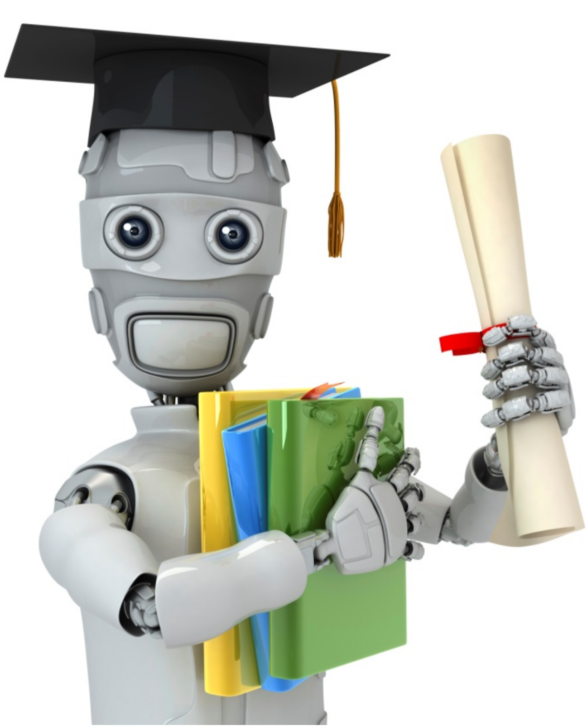
- Manufacturing
- Monitoring computers in a data center.
  - $x^{(i)}$ = features of machine i

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

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

    …

$p(x) < \varepsilon$



Machine Learning

Anomaly detection

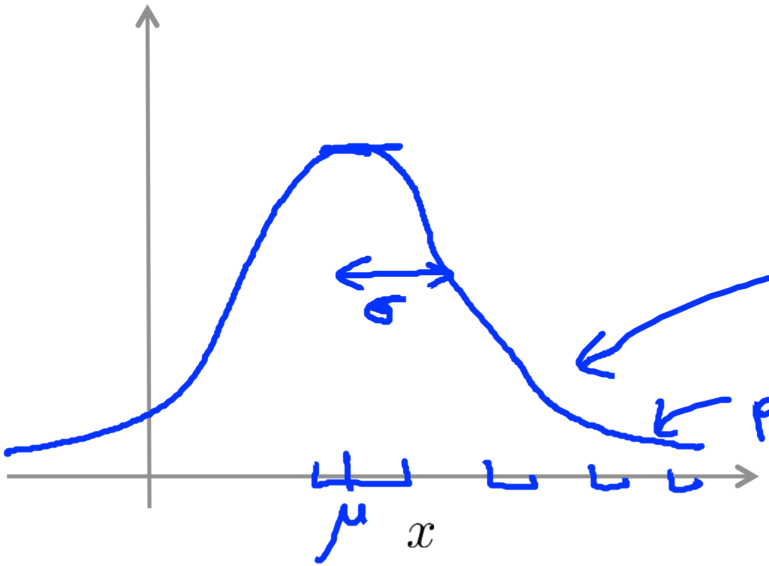
Gaussian distribution

Gaussian (Normal) distribution

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

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

σ standard deviation

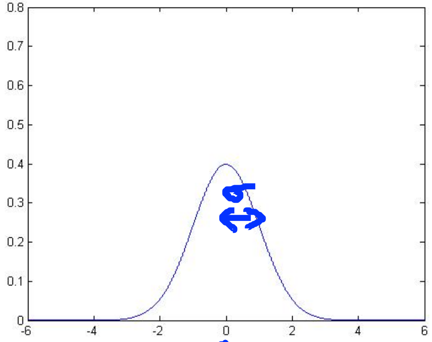


$p(x;\mu,\sigma^2)$

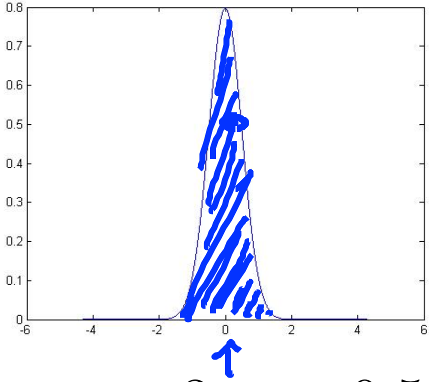
$$p(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Gaussian distribution example

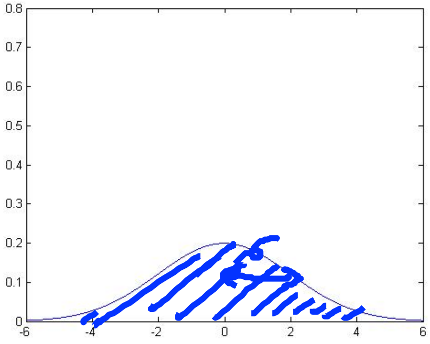
$\mu = 0, \sigma = 1$



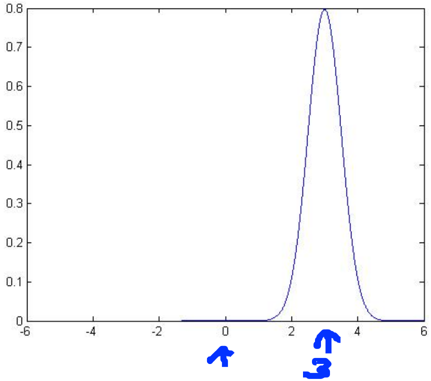
$\mu = 0, \sigma = \underline{0.5}$ $\quad \sigma^2 = 0.25$



$\mu = 0, \sigma = 2$



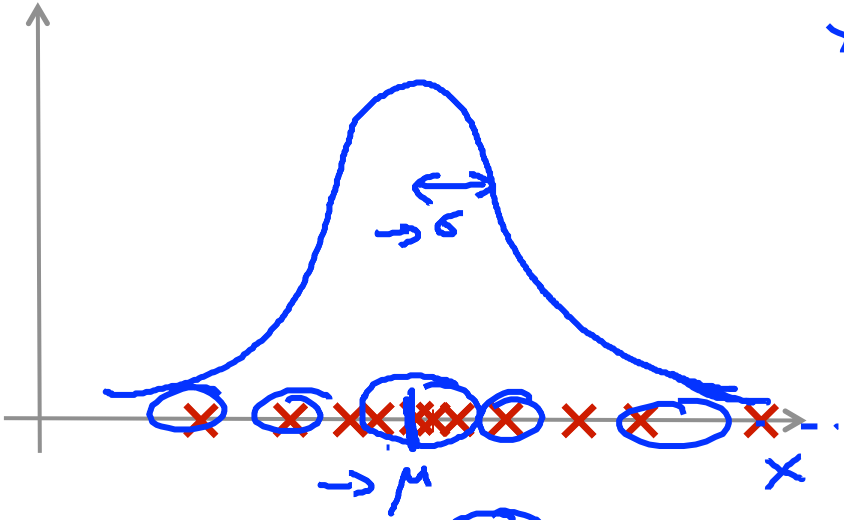
$\mu = 3, \sigma = 0.5$



Parameter estimation

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

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



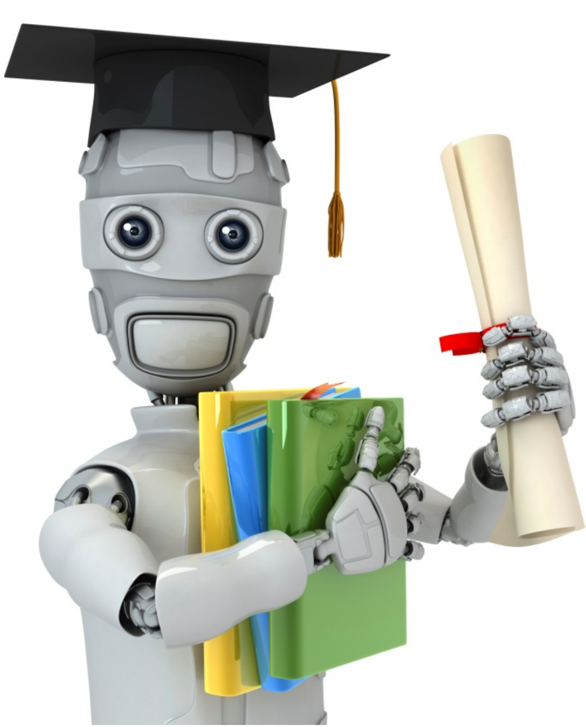
$$\mu = \frac{1}{m} \sum_{i=1}^{m} x^{(i)}$$

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

$m-1$

$\frac{1}{m-1}$ ← In statistics we use 1/m-1

Andrew Ng



Machine Learning

Anomaly detection

Algorithm

When dealing with a collection of data points, they will typically have a certain distribution (e.g. a Gaussian distribution). To detect anomalies in a more quantitative way, we first calculate the probability distribution p(x) from the data points. Then when a new example, x, comes in, we compare p(x) with a threshold r. If p(x)<r, it is considered as an anomaly. This is because normal examples tend to have a large p(x) while anomalous examples tend to have a small p(x)

Density estimation

Training set: $\{x^{(1)}, \ldots, x^{(m)}\}$
Each example is $x \in \mathbb{R}^n$

Assuming features are normally distributed

$x_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$
$x_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$
$x_3 \sim \mathcal{N}(\mu_3, \sigma_3^2)$

$\rightarrow p(x)$

Assuming independences

$$= \boxed{p(x_1; \mu_1, \sigma_1^2)\, p(x_2; \mu_2, \sigma_2^2)\, p(x_3; \mu_3, \sigma_3^2) \cdots p(x_n; \mu_n, \sigma_n^2)}$$

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

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

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

Anomaly detection algorithm

Average value of j feature

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

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

Vectorized

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

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

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

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

$\mu_1, \mu_2, \dots, \mu_n$

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

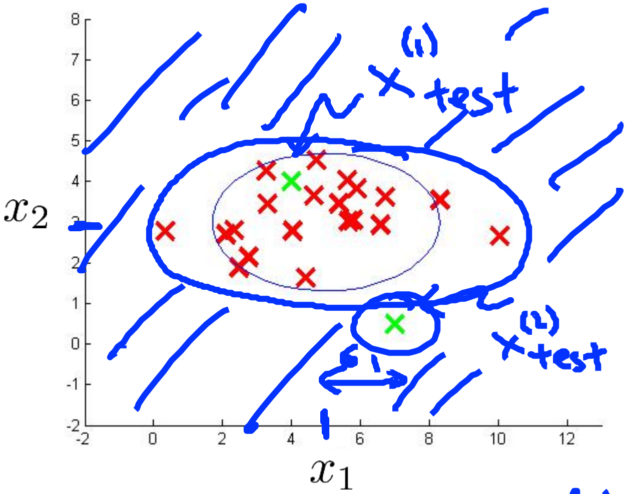
3. Given new example x, compute $p(x)$:

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

Anomaly if $p(x) < \varepsilon$

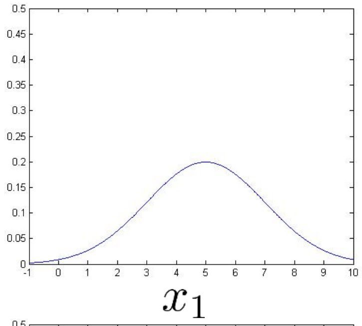
$\rightarrow p(x) = p(x_1; \mu_1, \sigma_1^2) \times p(x_2; \mu_2, \sigma_2^2)$

Anomaly detection example

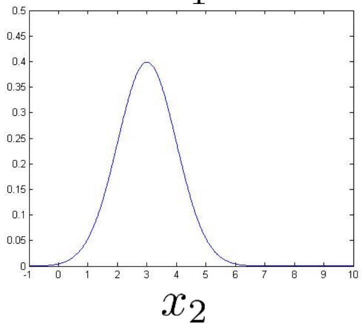


σ_1^2, σ_2^2 $= 4$

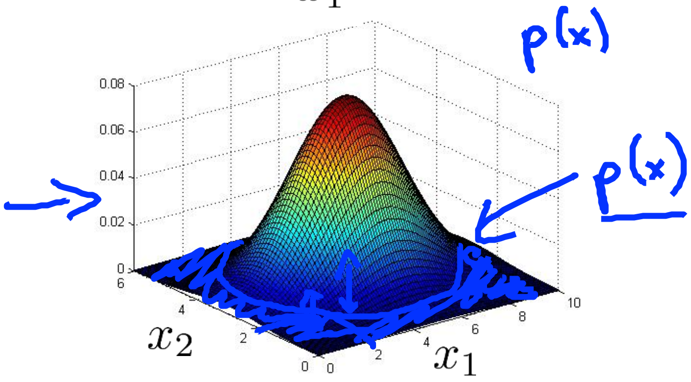
$$\boxed{\begin{aligned} \mu_1 &= 5, \sigma_1 = 2 \\ \mu_2 &= 3, \sigma_2 = 1 \end{aligned}}$$



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



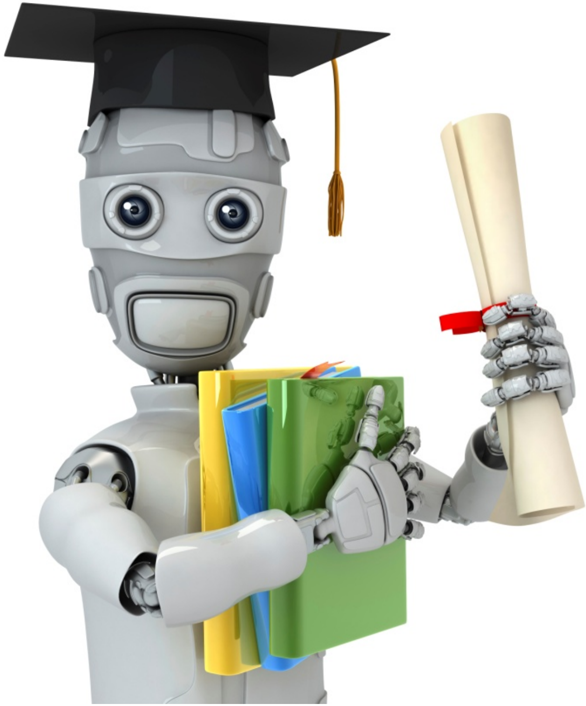
$p(x_2; \mu_2, \sigma_2^2)$



$$\varepsilon = 0.02$$

$$p(x_{test}^{(1)}) = 0.0426 \geq \varepsilon$$

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



Machine Learning

Anomaly detection

Developing and evaluating an anomaly detection system

The importance of real-number evaluation

Taking some features & leaving some, etc

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 labeled data, of anomalous and non-anomalous examples. ($y = 0$ if normal, $y = 1$ if anomalous).
- Training set: $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ (assume normal examples/not anomalous)
- Cross validation set: $(x_{cv}^{(1)}, y_{cv}^{(1)}), \ldots, (x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})})$
- Test set: $(x_{test}^{(1)}, y_{test}^{(1)}), \ldots, (x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$

We'll have in test set & cv set examples that are known to be anomalous

$y=1$

Aircraft engines motivating example

Un labeled training set

$\boxed{10000}$ good (normal) engines

$\boxed{20}$ flawed engines (anomalous) $\underline{2-50}$ $\underline{y=1}$

$\mu_1, \sigma_1^2, \ldots, \mu_n, \sigma_n^2$

Training set: $\boxed{6000}$ good engines $(y=0)$ $p(x) = p(x_1; \mu_1, \sigma_1^2) \cdots p(x_n; \mu_n, \sigma_n^2)$

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

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

Alternative: way of splitting data set

Training set: 6000 good engines

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

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

Same 4k sets are used for cv & test

Andrew Ng

Algorithm evaluation

- Fit model $p(x)$ on training set $\{x^{(1)}, \ldots, x^{(m)}\}$
- On a cross validation/test example x, predict

$(x_{test}^{(i)}, y_{test}^{(i)})$

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

Skewed case

$y = 0$

Possible evaluation metrics:

- - True positive, false positive, false negative, true negative
- - Precision/Recall
- - F_1-score

Accuracy isn't enough due to skewed case

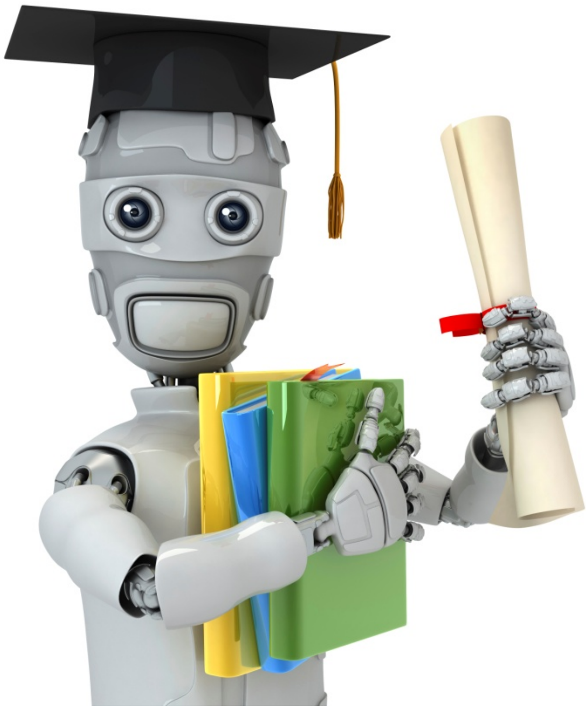
CV

Test set

Try diff. values of epsilon that maximizes the evaluation metric

Can also use cross validation set to choose parameter ε

Andrew Ng



Machine Learning

Anomaly detection

Anomaly detection vs. supervised learning

Anomaly detection vs. Supervised learning

- Very small number of positive examples ($y = 1$). (0-20 is common).
- Large number of negative ($y = 0$) examples. $\boxed{p(x)}$
- Many different "types" of anomalies. Hard for any algorithm to learn from positive examples what the anomalies look like;
- future anomalies may look nothing like any of the anomalous examples we've seen so far.

Large number of positive and negative examples.

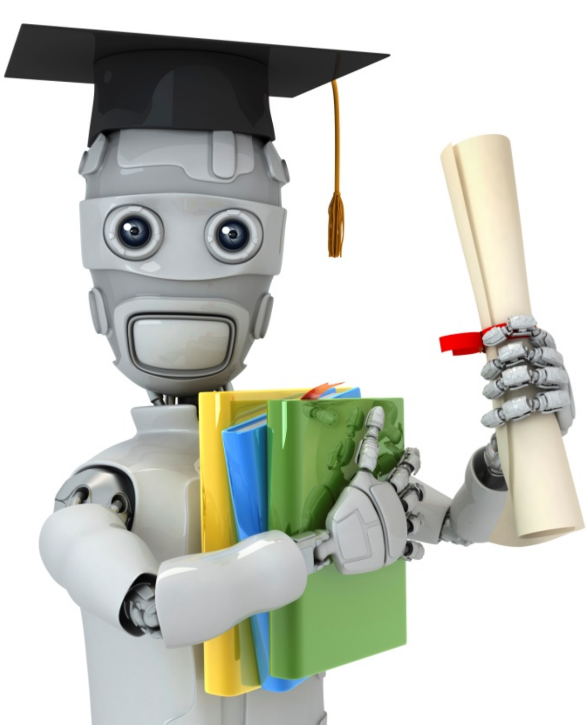
Like having many diff. types of spam emails so we've many diff. types of anomalies

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

Anomaly detection vs. Supervised learning

Anomaly detection	Supervised learning
Fraud detection $y=1$	Email spam classification
Manufacturing (e.g. aircraft engines)	Weather prediction (sunny/rainy/etc).
Monitoring machines in a data center	Cancer classification
⋮	⋮



Machine Learning

Anomaly detection

Choosing what features to use

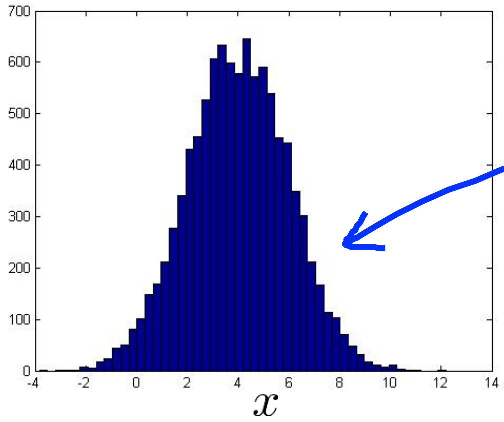
Non-gaussian features

Doing some transformations to the data to look more like gaussian distr.

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

Some feature x1

hist



$x_1 \leftarrow \log(x_1)$

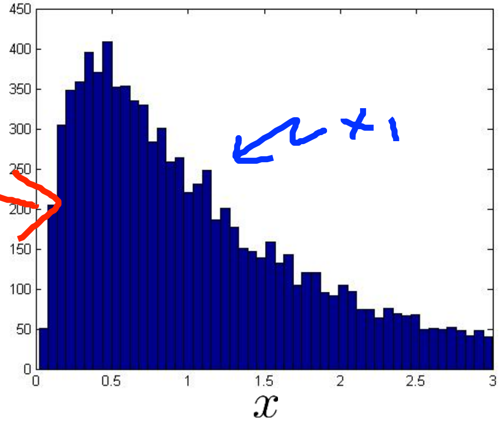
$x_2 \leftarrow \log(x_2 + 1)$

$\log(x_2 + c)$

Const

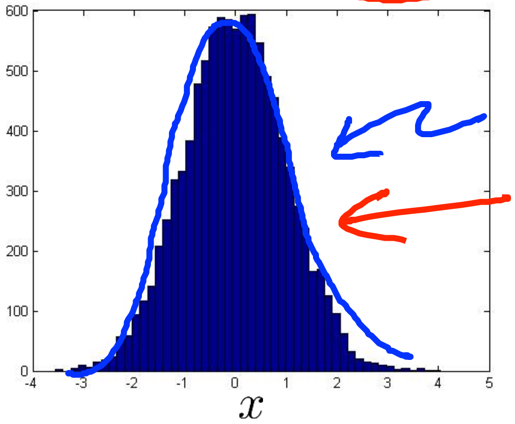
$x_3 \leftarrow \sqrt{x_3} = x_3^{\frac{1}{2}}$

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



x_1

$\xrightarrow{\log(x)}$



Error analysis for anomaly detection

Creating new feature for anomaly detection

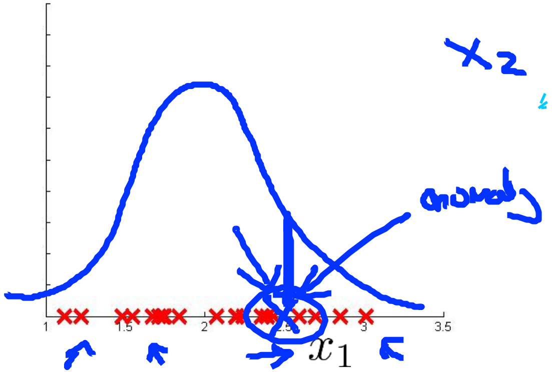
Want $p(x)$ large for normal examples x.

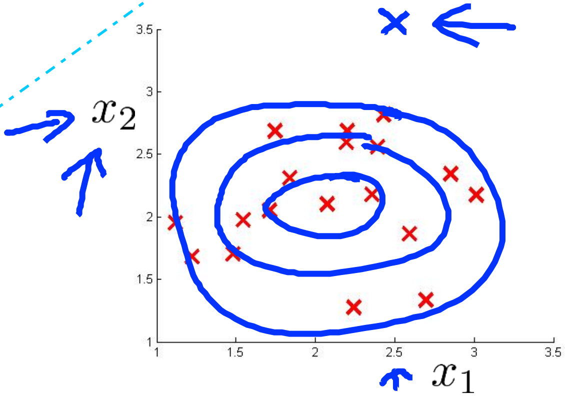
$p(x)$ small for anomalous examples x.

Look at the anomalies & try to create some new features to become easier to distinguish anomalies

Most common problem:

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





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 = number of disk accesses/sec

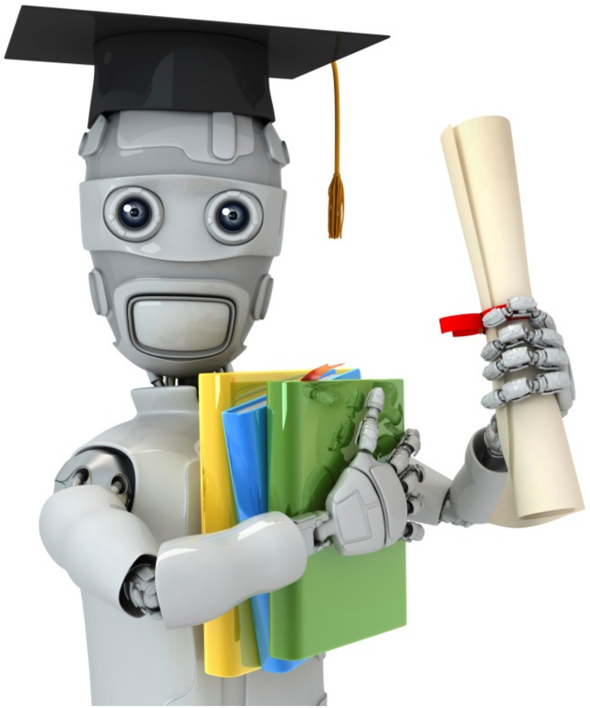
x_3 = CPU load

x_4 = network traffic

Created new feature

$$x_5 = \frac{\text{CPU load}}{\text{network traffic}} \qquad x_6 = \frac{(\text{CPU load})^2}{\text{network traffic}}$$

Try coming up with more features to distinguish between the normal and the anomalous examples if Anomaly detection algorithm is performing poorly and outputs a large value of p(x) for many normal examples and for many anomalous examples in your cross validation dataset.

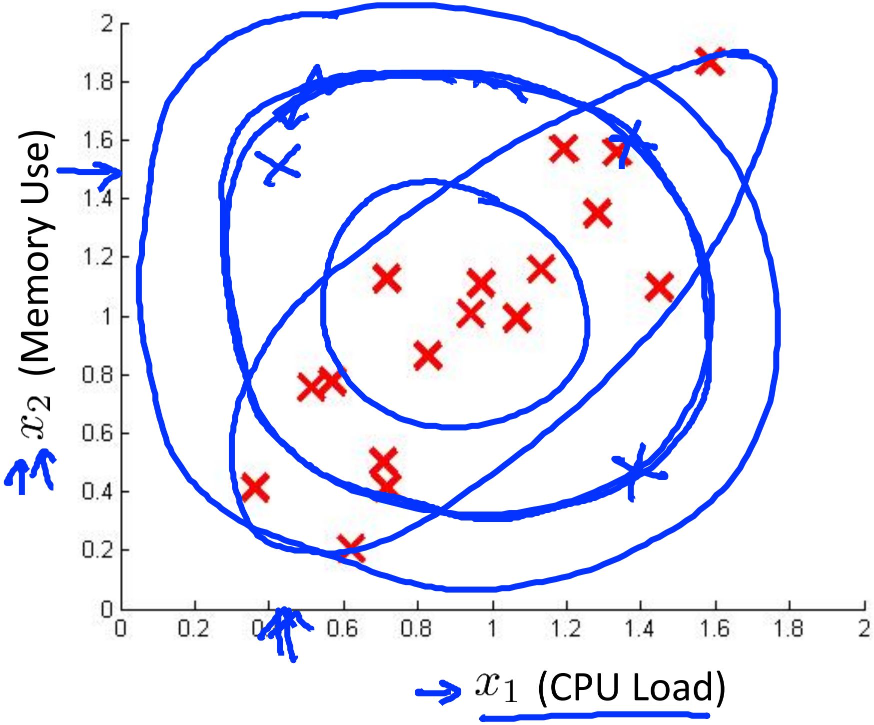


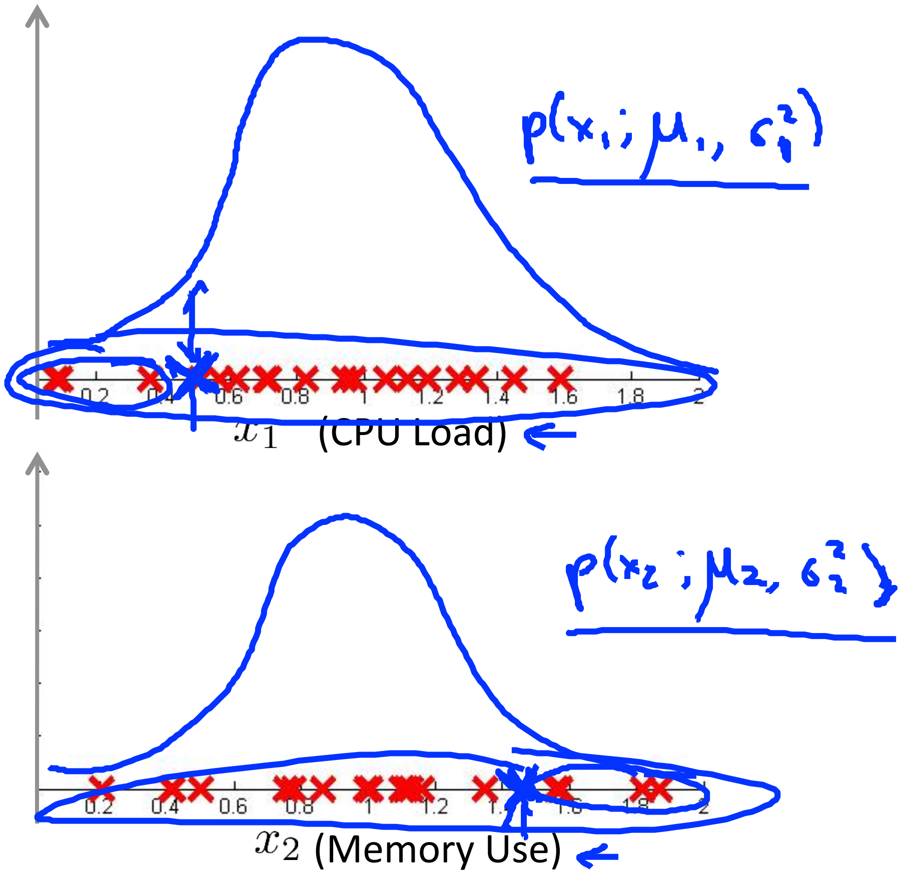
Machine Learning

Anomaly detection

Multivariate Gaussian distribution

Motivating example: Monitoring machines in a data center





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

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

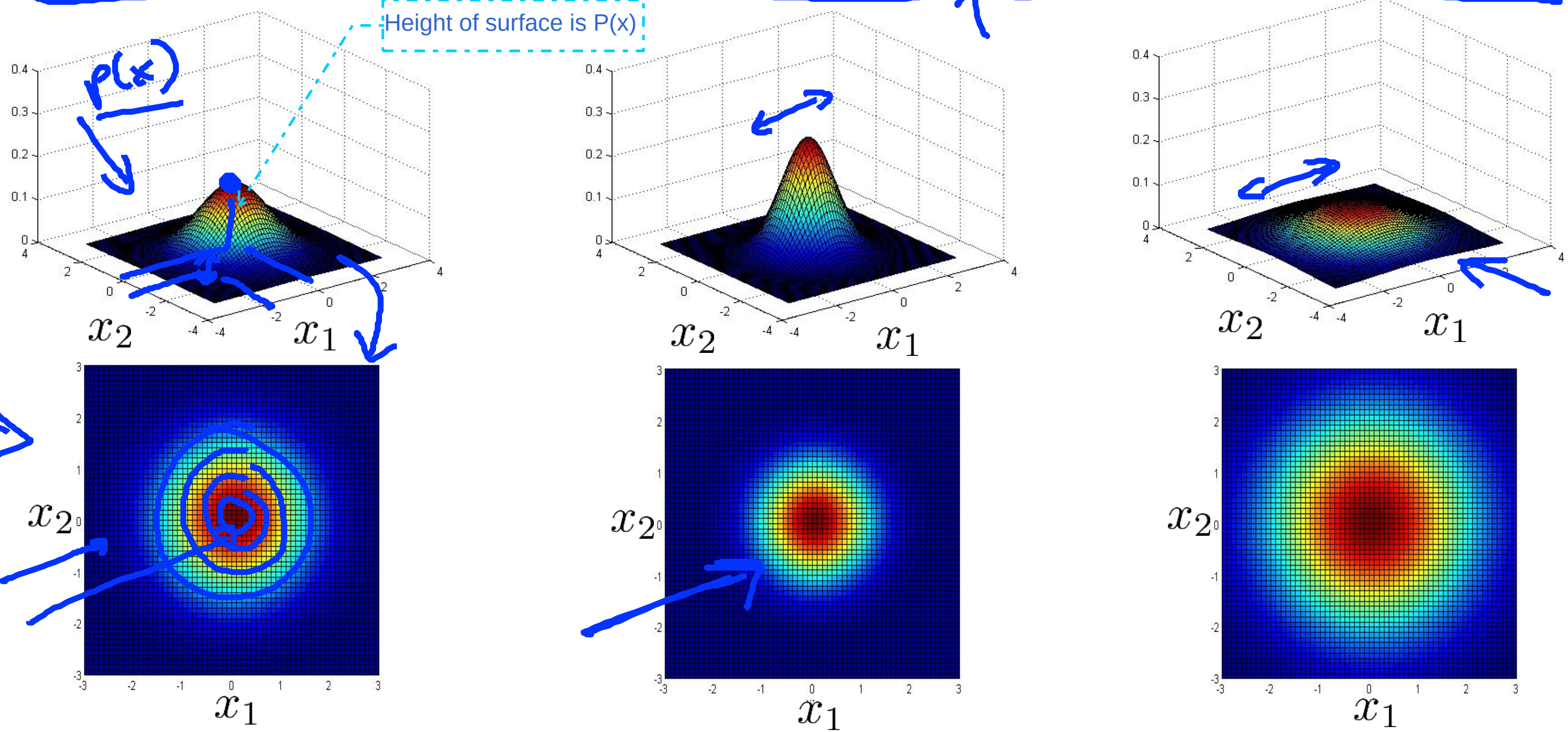
$|\Sigma|$ = determinant of Σ | det(Sigma)

Multivariate Gaussian (Normal) examples

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

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

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

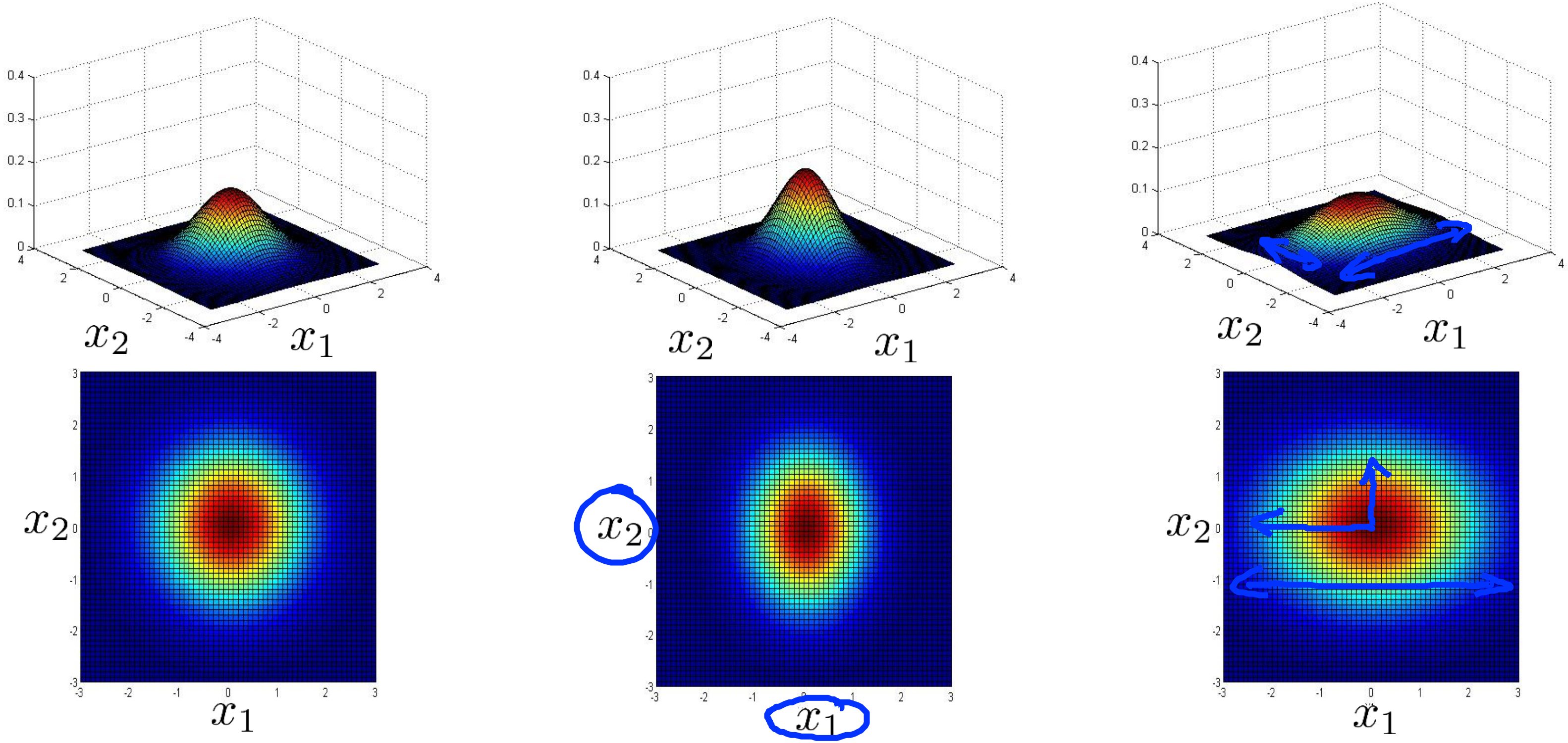


Multivariate Gaussian (Normal) examples

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

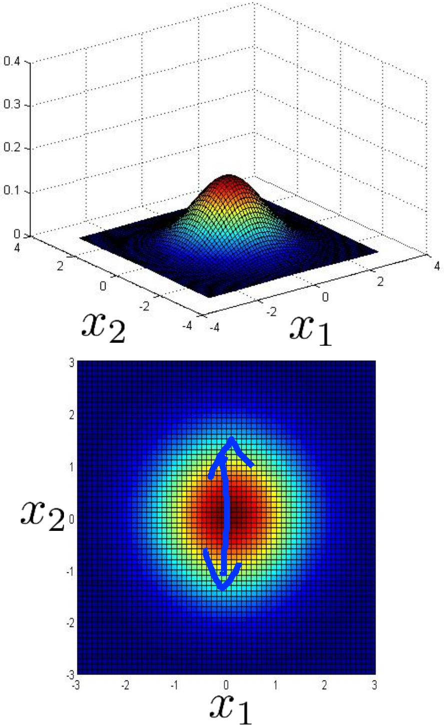
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 1 \end{bmatrix}$$

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

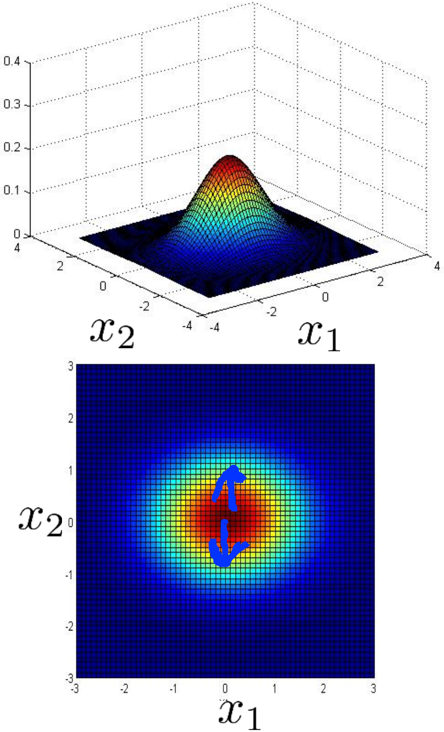


Multivariate Gaussian (Normal) examples

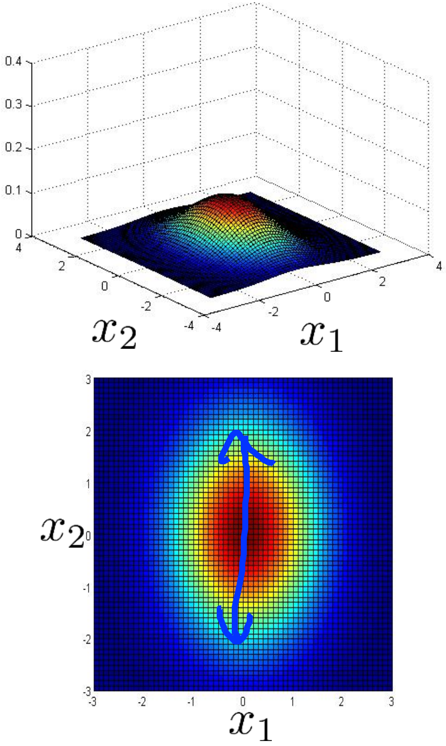
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



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

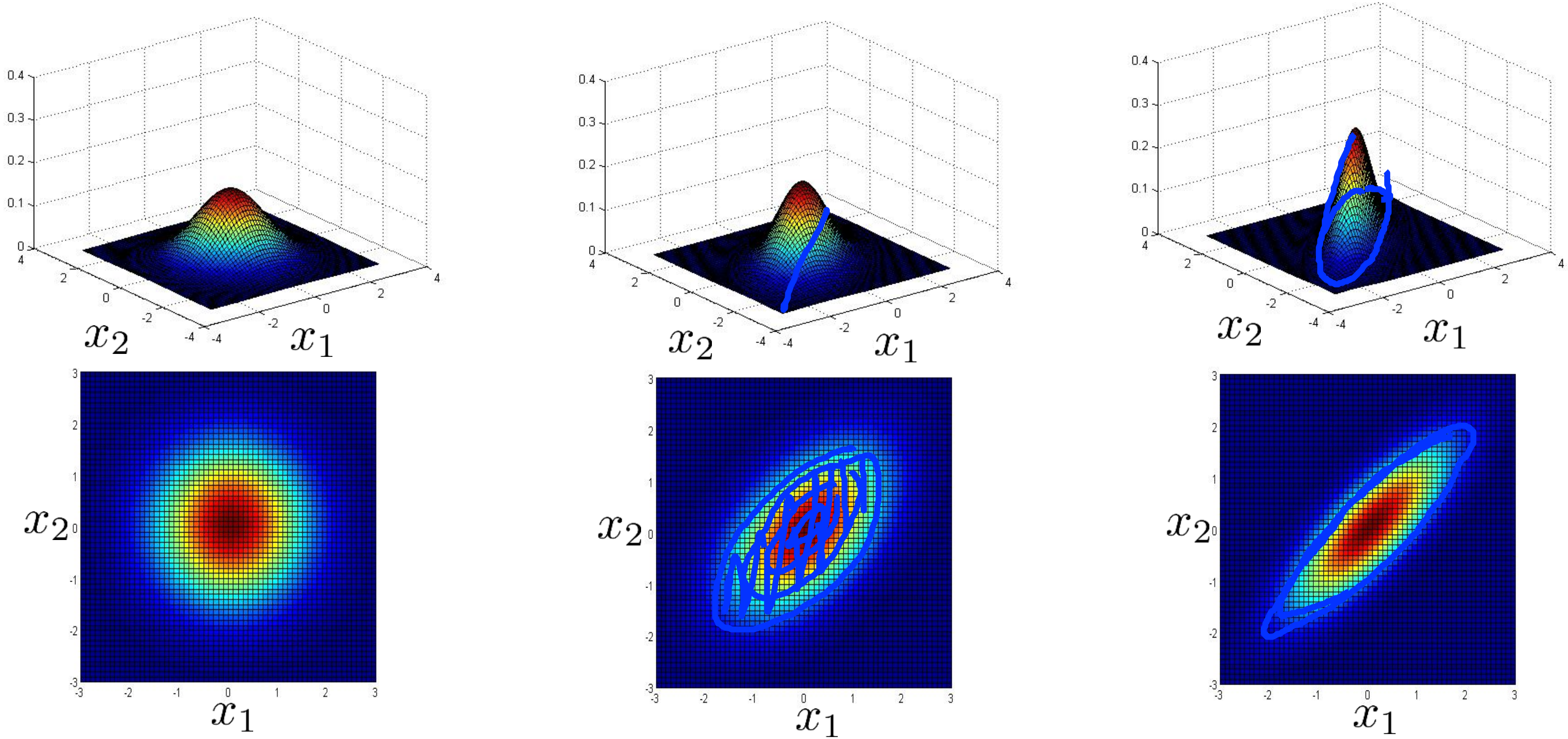


$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$



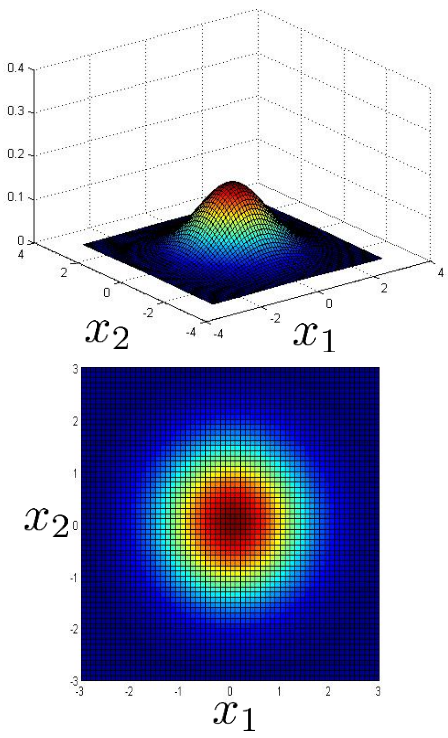
Multivariate Gaussian (Normal) examples

$$\mu = \begin{bmatrix}0\\0\end{bmatrix} \; \Sigma = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} \qquad \mu = \begin{bmatrix}0\\0\end{bmatrix} \; \Sigma = \begin{bmatrix}1 & 0.5\\0.5 & 1\end{bmatrix} \qquad \mu = \begin{bmatrix}0\\0\end{bmatrix} \; \Sigma = \begin{bmatrix}1 & 0.8\\0.8 & 1\end{bmatrix}$$

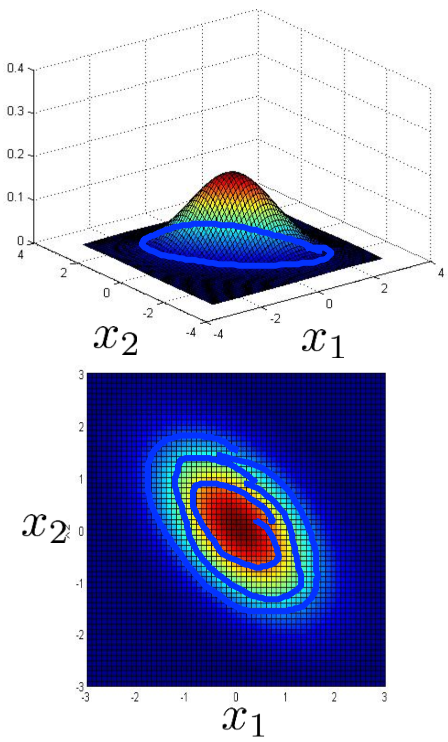


Multivariate Gaussian (Normal) examples

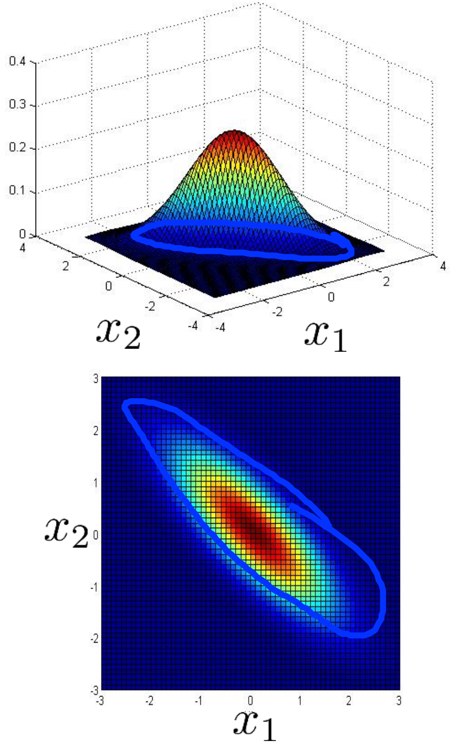
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



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



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



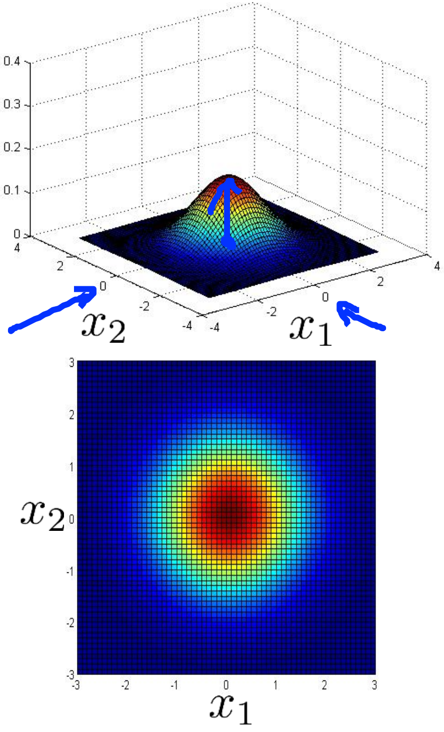
Andrew Ng

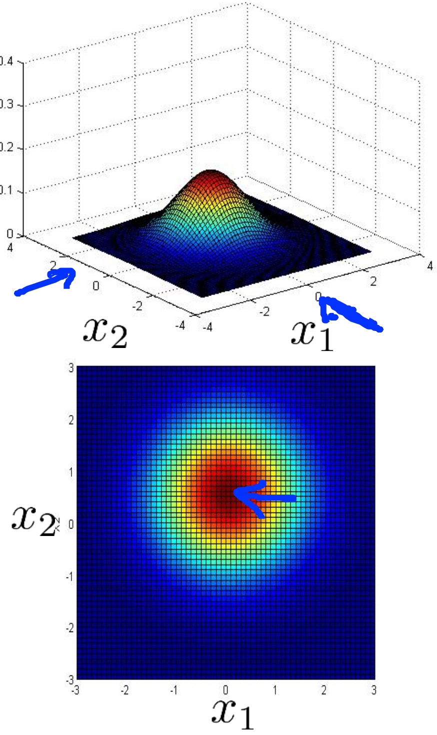
Multivariate Gaussian (Normal) examples

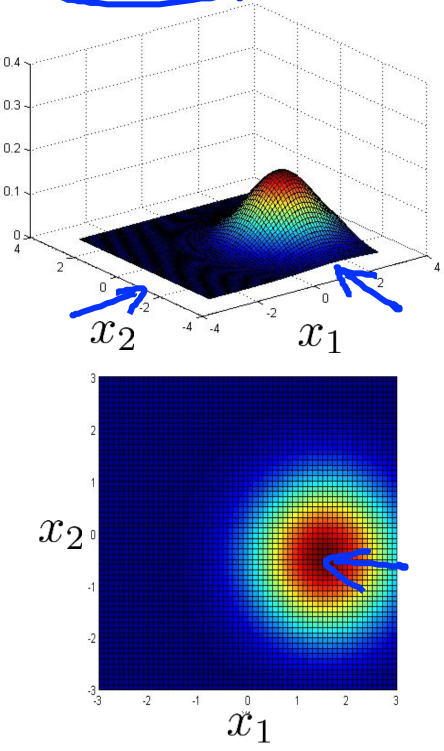
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

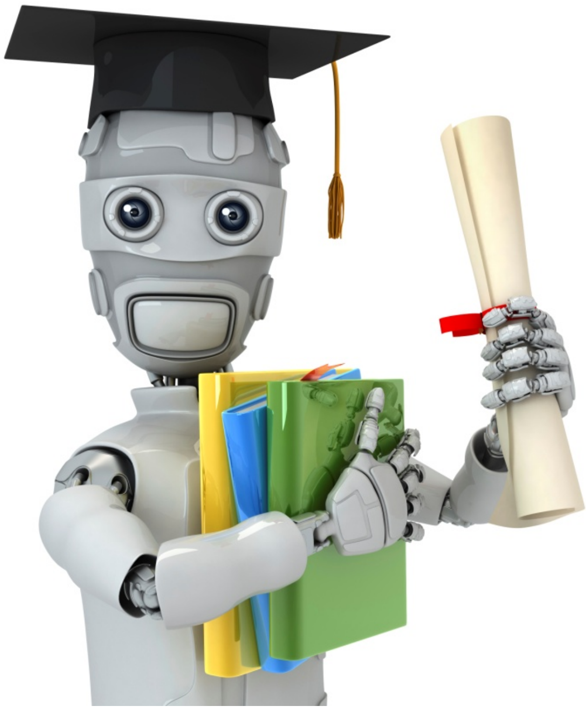
$$\mu = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

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









Machine Learning

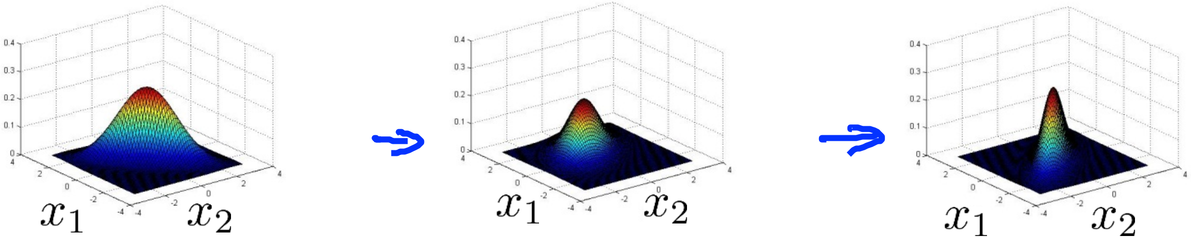
Anomaly detection

Anomaly detection using the multivariate Gaussian distribution

Multivariate Gaussian (Normal) distribution

Parameters μ, Σ $\qquad \mu \in \mathbb{R}^n \qquad \Sigma \in \mathbb{R}^{n \times n}$

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



Parameter fitting:

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

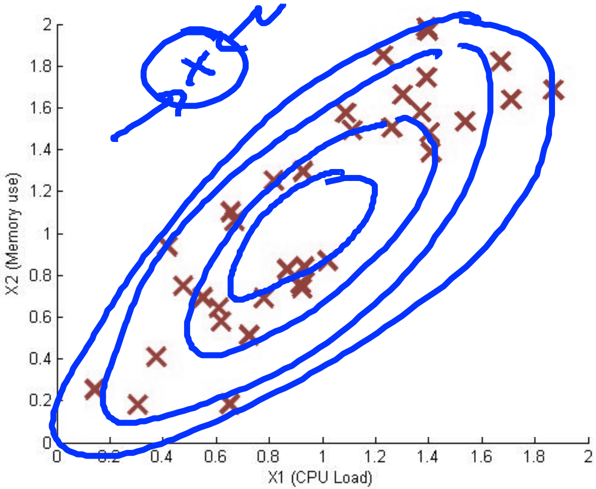
$$\boxed{\mu} = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} \qquad \boxed{\Sigma} = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)}-\mu)(x^{(i)}-\mu)^T$$

Anomaly detection with the multivariate Gaussian

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

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

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



2. Given a new example x, compute

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

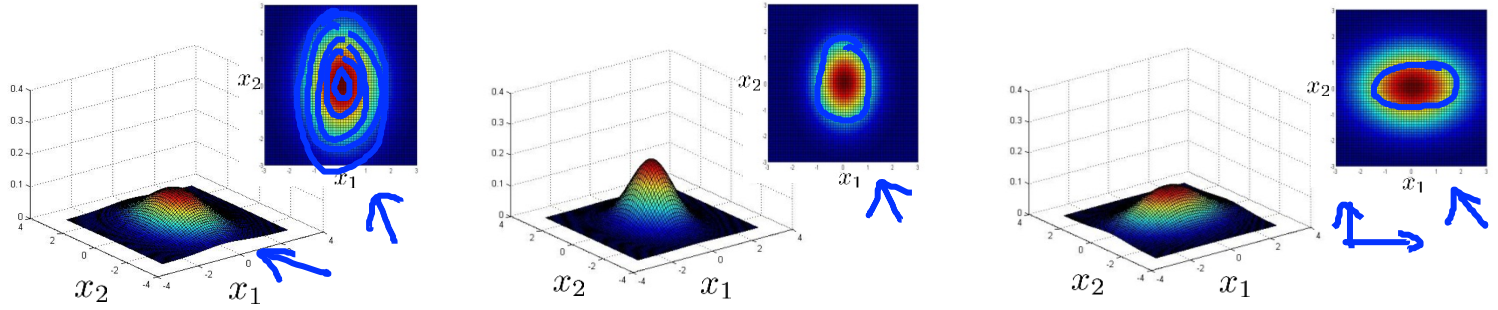
Flag an anomaly if $p(x) < \varepsilon$

Relationship to original model

Special model of mulivariate gauss. distribution but with a constraint as the PDF is axis aligned

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

p(x)



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 $\Sigma =$

This is the constraint

Andrew Ng

Original model vs. Multivariate Gaussian

Used often

$$p(x_1;\mu_1,\sigma_1^2)\times\cdots\times p(x_n;\mu_n,\sigma_n^2)$$

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

Manually create features to capture anomalies where x_1, x_2 take unusual combinations of values.

$$\boxed{x_3=\frac{x_1}{x_2}=\frac{\text{CPU load}}{\text{memory}}}$$

Automatically captures correlations between features

If sigma is found to be non invertible so check for redundant features & m > n

$\Sigma \in \mathbb{R}^{n\times n}$ Σ^{-1}

Computationally cheaper (alternatively, scales better to large m)

$n=10{,}000$, $n=100{,}000$

Computationally more expensive

Multivariate gauss. is considered if this condition is satisfied

$\rightarrow \Sigma \sim \frac{n^2}{2}$

$x_1 = \cancel{x_2}$

$\cancel{x_3} = x_4 + x_5$

OK even if m (training set size) is small

Must have $\boxed{m>n}$ or else Σ is non-invertible.

$\boxed{m \geq 10n}$