ANOMALY DETECTION

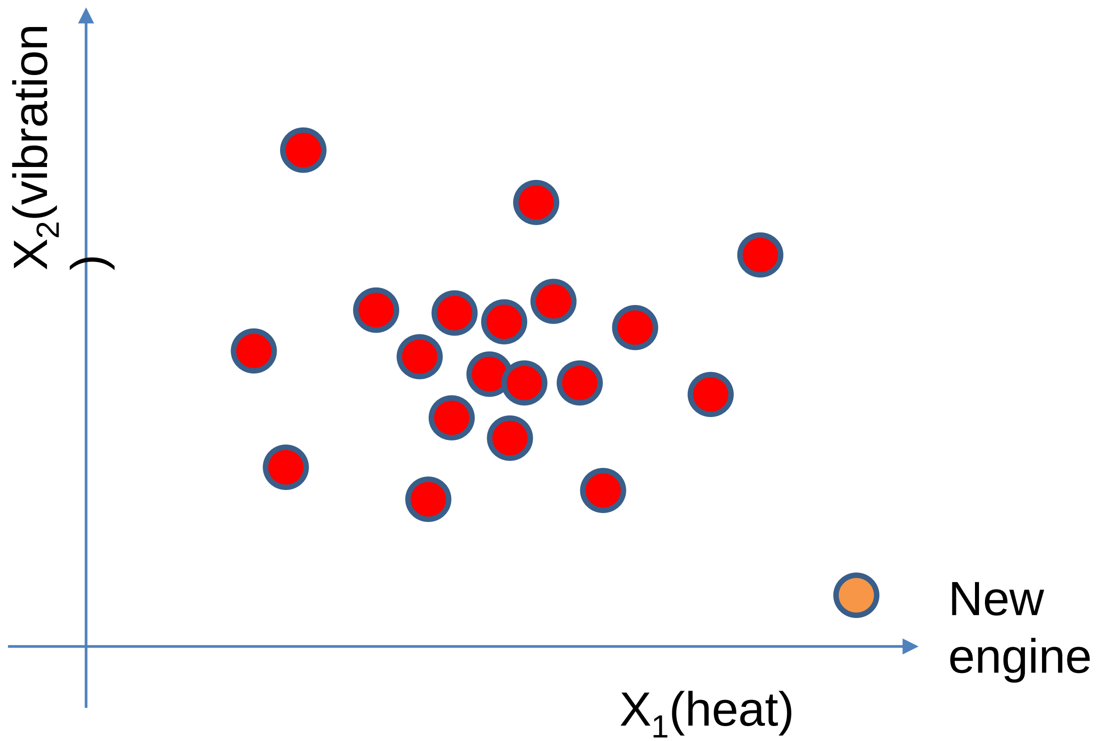
References

- Anomaly Detection – A Survey PDF

- A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data PDF

What is Anomaly Detection?

- Example
  - Suppose you are a car manufacturer, and perform car engine testing.
  - You are given some features
    - X_1 = heat generated
    - X_2 = vibration intensity
  - Given DataSet: $x^{(1)}, x^{(2)}, \dots, x^{(n)}$
  - Then for a new engine
    - We want to determine if x_{test} is an anomaly

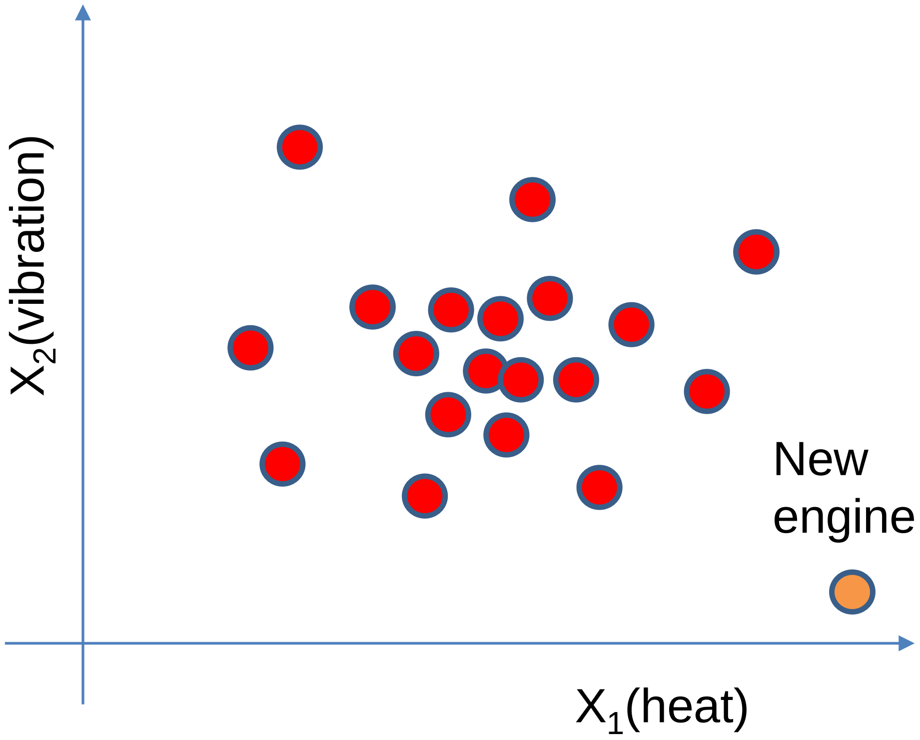


What is Anomaly Detection

- Anomaly detection identifies unexpected items or events in datasets, which differ from the norm.

- Usually anomaly detection is applied to unlabeled data (so its not a standard classification task).

What is Anomaly Detection



- Given a set of n ‘normal’ data points, we build a model for p(x).
- Then given a new data point, if
  - $p(x) < \varepsilon$ ---> anomaly
  - $p(x) \geq \varepsilon$ ----> normal

Other Applications of Anomaly Detection

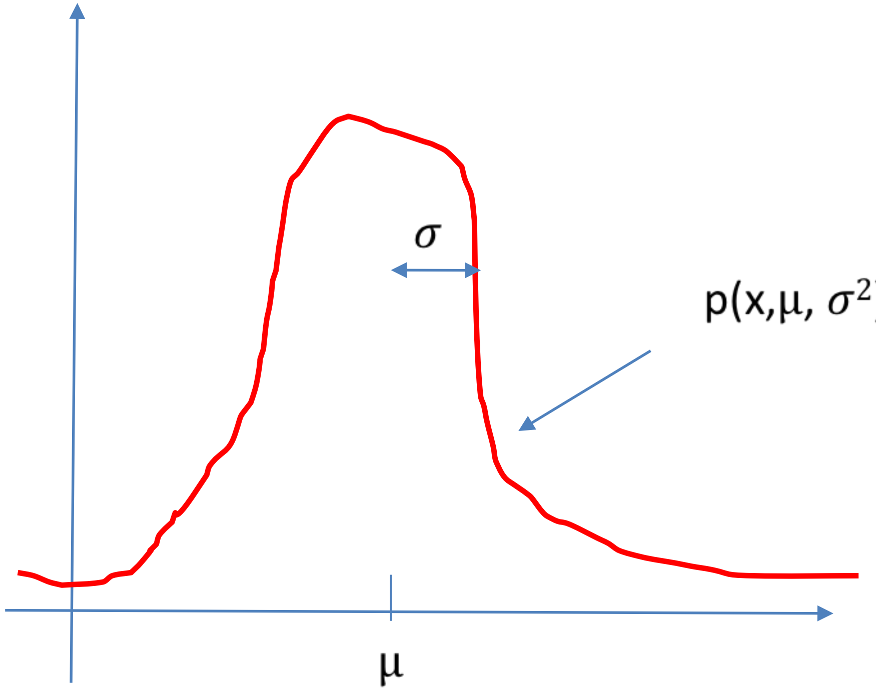
- Fraud detection
  - $X^{(i)}$ = features of a users i's activities
  - Model p(x) from data
  - Identify unusual users by checking which have $p(x) < \varepsilon$

X_1 => login frequency
X_2 => number of transactions
X_3 => typing speed

- Monitoring computers in a data center
  - $X^{(i)}$ = features of machine i
  - Features :
    - x_1 = memory use,
    - x_2 = number of disk accesses,
    - x_3 = CPU load,
    - x_4 = CPU load / network traffic
  - Model p(x) and then identify abnormal machines that may indicate bad nodes in the network

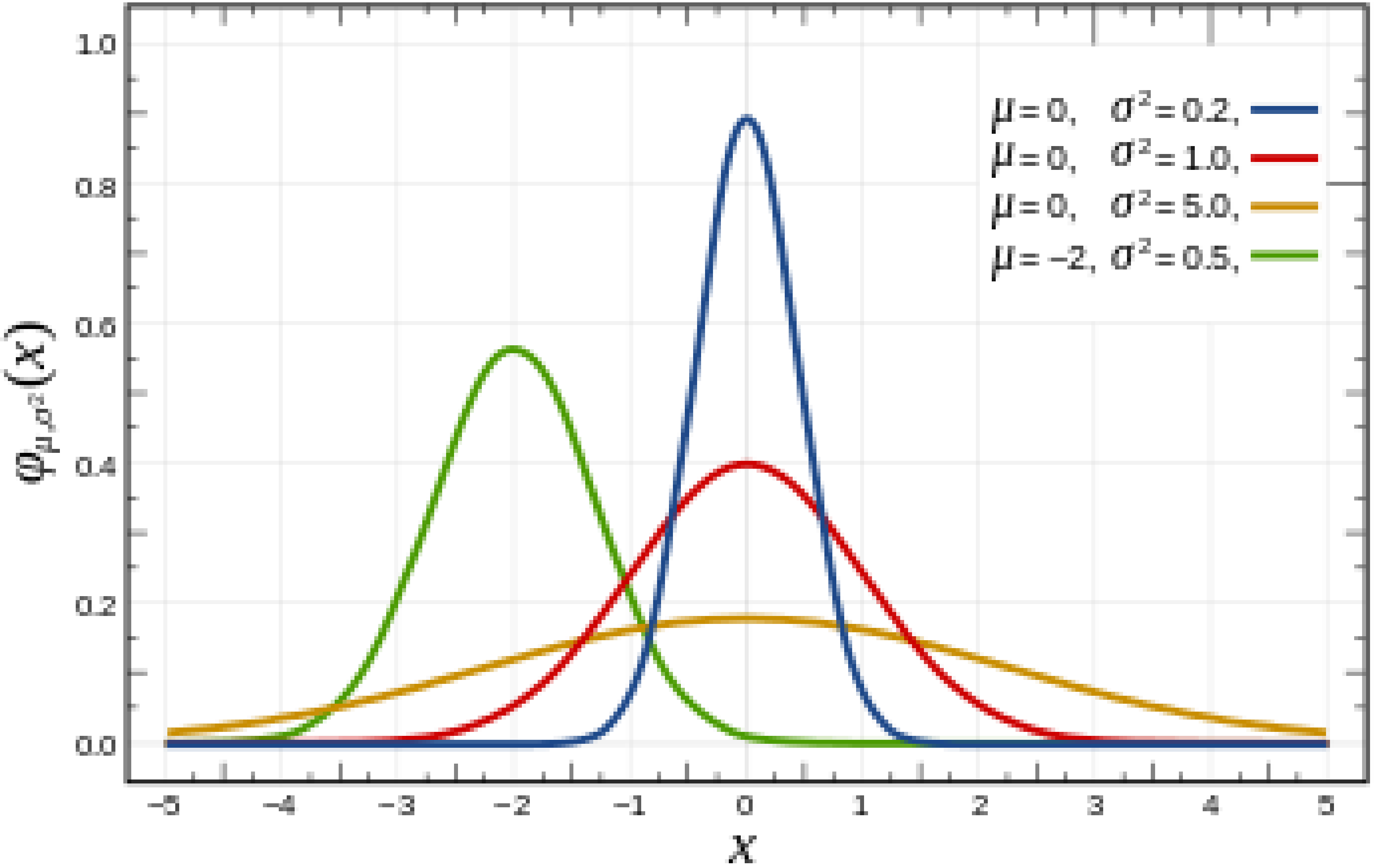
Gaussian (Normal) Distribution

- Say $x \in \mathbb{R}$.
- If x is a distributed Gaussian with mean μ, *variance* σ^2.
- $X \sim \aleph(\mu, \sigma^2)$



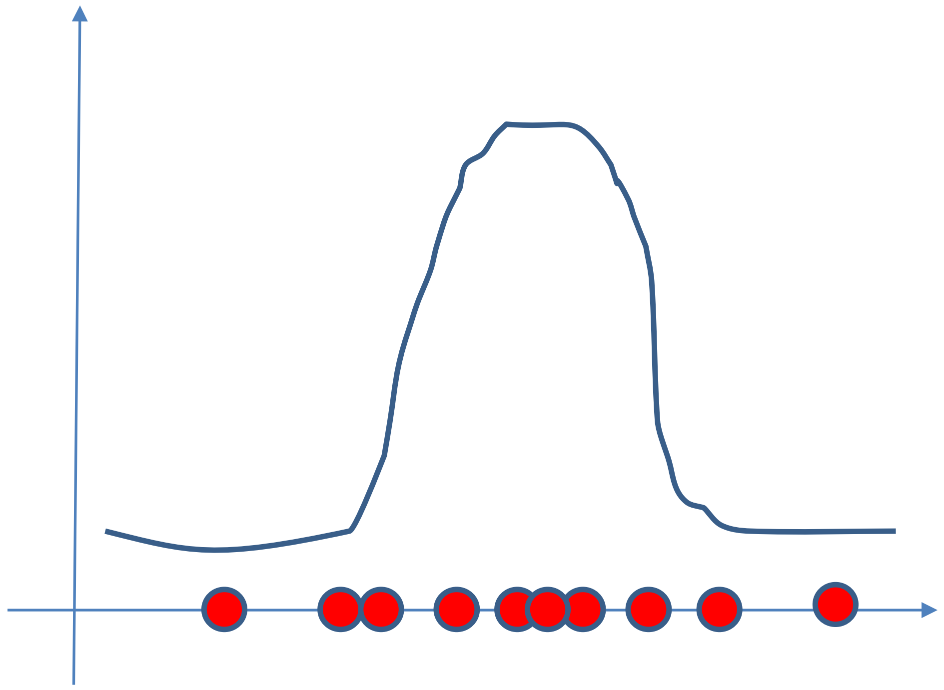
$$p(x,\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Gaussian Distribution Examples



Parameter Estimation

- Dataset: $\{ x^{(1)}, x^{(2)}, \ldots., x^{(m)} \}$ $x^{(i)} \varepsilon \mathbb{R}$



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

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

Density Estimation

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

- Model the probability of x
  - $p(x) = p(x_{1;}\, \mu_{1;}\, \sigma^2{}_1)p(x_{2;}\, \mu_{2;}\, \sigma^2{}_2)\, p(x_{3;}\, \mu_{3;}\, \sigma^2{}_3) \ldots p(x_{n;}\, \mu_{n;}\, \sigma^2{}_n) = \prod_{j=1}^{n} p(x_j\,;\, \mu_j, \sigma^2{}_j)$

- How to model these terms?
  - Assume that each feature follows a Gaussian distribution with some mean and variation.
  - So, for example assume $x_1 \sim \aleph\, (\mu_1, \sigma^2{}_1)$

Anomaly Detection Algorithm

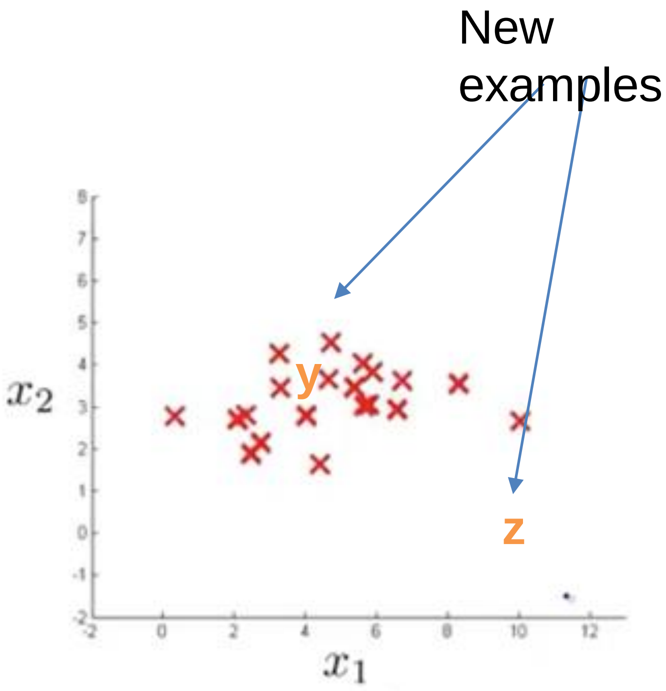
- Given a dataset of examples, { $x^{(1)}$, $x^{(2)}$, …. , $x^{(m)}$ }
- Choose features x_i that you think might be indicative of anomalous examples.
- Fit parameters $\mu_1, \ldots, \mu_n$; $\sigma^2{}_1$, … , $\sigma^2{}_n$
  - $u_j = \frac{1}{m}\sum_{i=1}^{m} x_j{}^{(i)}$
  - $\sigma^2{}_j = \frac{1}{m}\sum_{i=1}^{m}(x_j{}^{(i)} - \mu_j)^2$

- Given new example x, computer p(x):

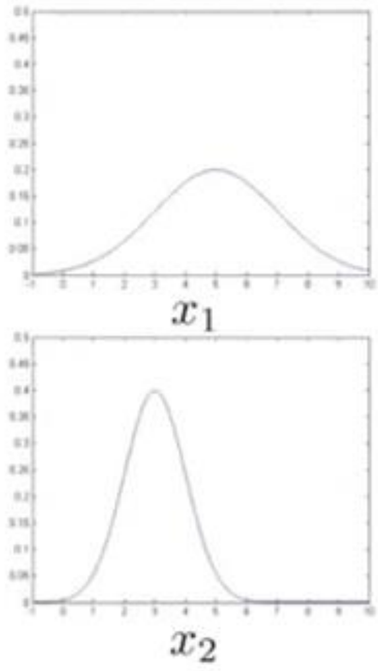
  - $p(x) = \prod_{j=1}^{n} p(xj\ ; \mu_j, \sigma^2{}_j) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_j}\ \exp(-\frac{(x_j - \mu_j)^2}{2\sigma^2{}_j})$

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

Example

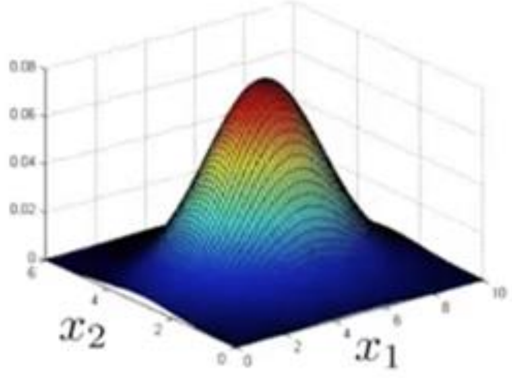


$$\mu_1 = 5, \sigma_1 = 2$$
$$\mu_2 = 3, \sigma_2 = 1$$



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

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



p(x) = p(x^1; μ_1, $\sigma^2{}_1$) * p(x_2; μ_2, $\sigma^2{}_2$)

$\varepsilon = 0.02$

p(y) = 0.0426 > ε

p(z) = 0.0021 < ε

How to evaluate the algorithm?

- We have been treating anomaly detection as an unsupervised learning problem, using non-labeled dataset.

- However, if we have some labeled data, of anomalous and non-anomalous examples (y=0 if normal, y=1 if anomalous), then we can use this information to make decisions about our learning algorithm.
  - Such as which features to choose

- Training set: $x^{(1)}$, $x^{(2)}$, … $x^{(n)}$ assume normal examples / not anamalous
- Cross Validation set : normal and anamalous examples
- Test Set : normal and anamalous examples

Example (1/2)

- Dataset
  - 1000 good (normal) examples
  - 20 flawed examples (anomalous)

- Splitting Dataset

  y=0 $p(x) = p(x_1, \mu_1, \sigma_1^2) * p(x_2, \mu_2, \sigma_2^2)$ ...

  - **Training set (60%) :** 6000 good examples (unlabeled training set)
  - **Cross validation set(20%) :** 2000 good examples (y=0) & 10 anomalous (y=1)
  - **Test set(20%) :** 2000 good examples (y=0) & 10 anomalous (y=1)

Example (2/2)

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

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

- Possible evaluation metrics:
  - True positive, false positive, false negative, true negative
  - Precision / recall
  - F_1-recall
- Can you also use cross validation set to choose parameter ε

Anomaly Detection VS Supervised Learning

- Very small number of positive examples (y=1) maybe 0 – 20 anomalous examples.
- Large number of negative (y=0) examples
- Many different “types” of anomalies. Hard for any algorithm to learn from positive examples what future anomalies may look like

- 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.

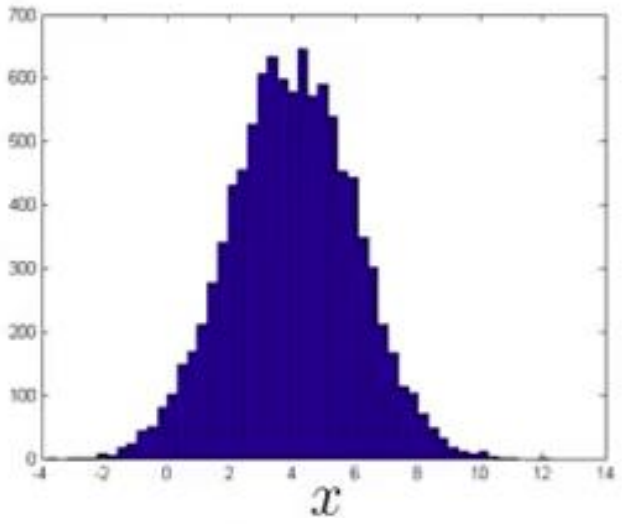
Anomaly Detection

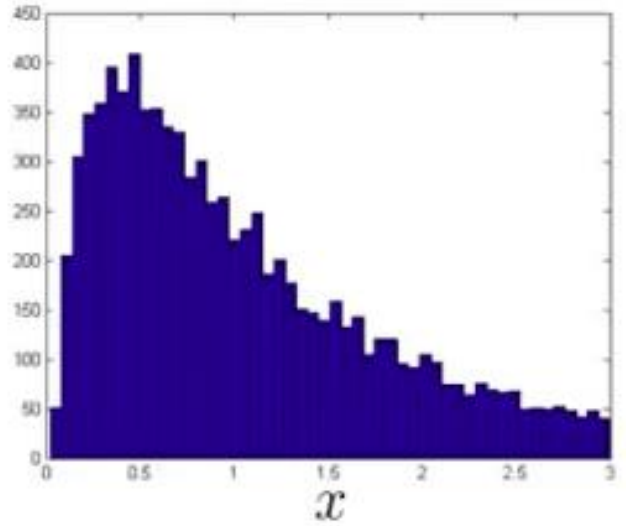
- Fraud Detection
- Manufacturing
- Monitoring machines in a data center

Supervised Learning

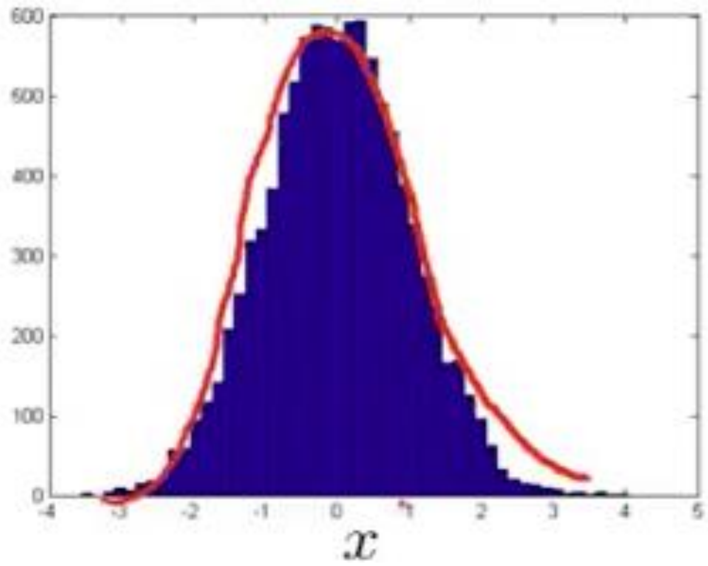
- Email Spam Classification
- Weather prediction
- Cancer classification

Non-gaussian Features





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
- Solution
  - Find new features that expose the difference between anomalous and normal examples

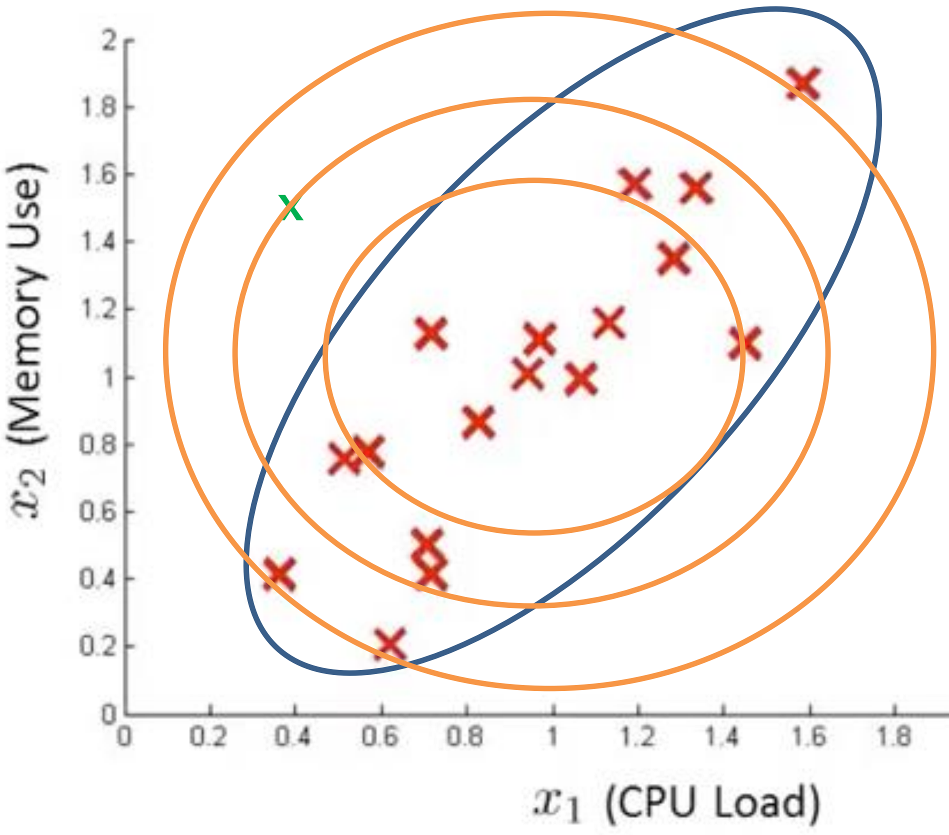
Deriving features

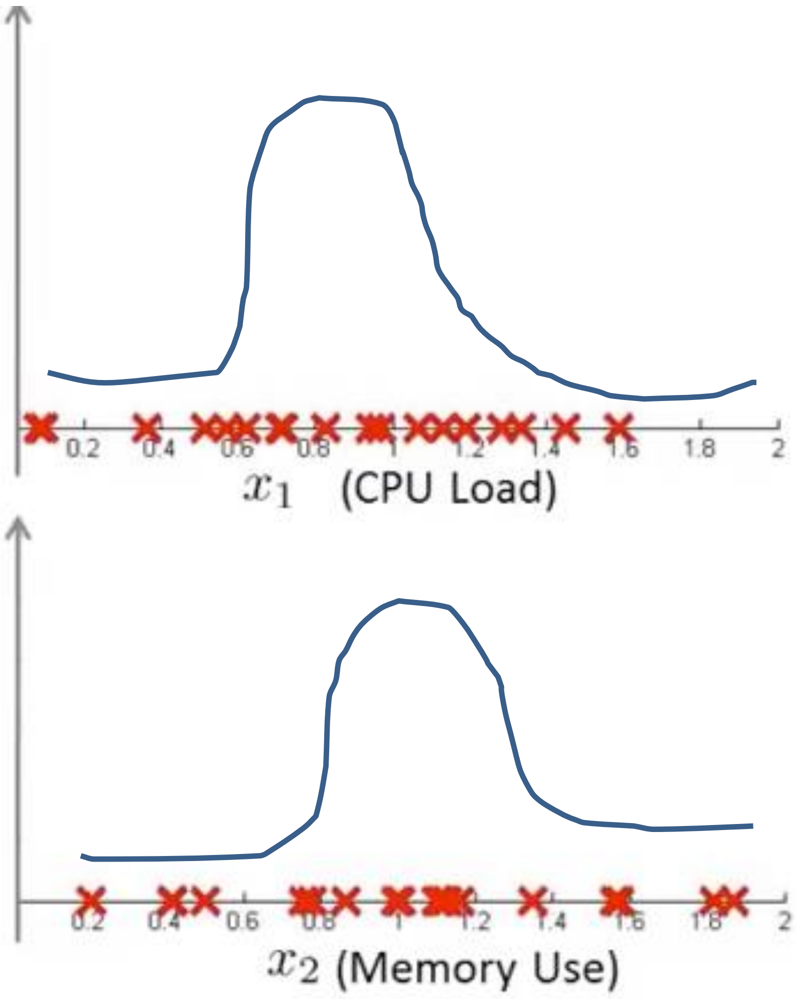
- Consider the following example
  - Lets assume we want to detect anomalies in a data center
  - We are given a set of features
    - X_1 = memory use of computer
    - X_2 = number of disk accesses / sec
    - X_3 = CPU load
    - X_4 – network traffic

  - The goal is to choose features that might take on unusually large or small values in the event of an anomaly.
    - May derive features such as
      - x_5 = cpu load / network traffic

Ended Here

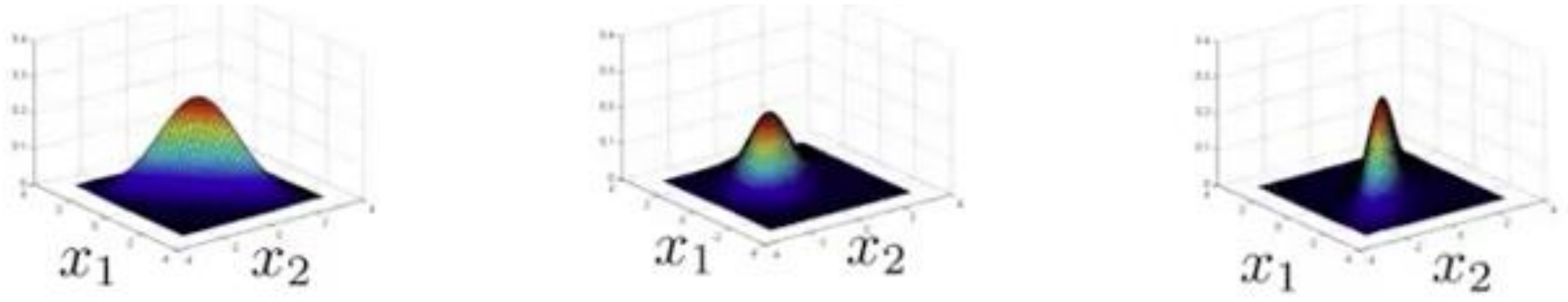
Multivariate Gaussian : Motivating Example





Multivariate Gaussian (Normal) Distribution

- $X \in \mathbb{R}^n$, don't model $p(x_1), p(x_2), \ldots, p(x_n)$. separately.
- Model p(x) collectively.
- Parameters : $\mu \in \mathbb{R}^n , \Sigma \in \mathbb{R}^{n \times n}$ $(covariance\ matrix)$
- $p(x; \mu ; \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(- \frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$



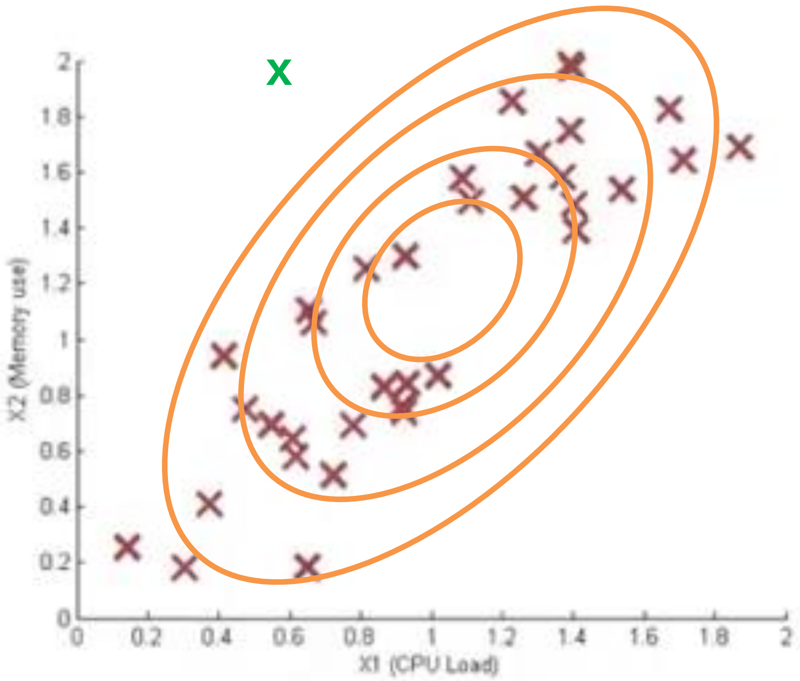
We get different distributions as we vary these parameters

Parameter fitting

- Given a training set { x$^{(1)}$, x $^{(2)}$, …. , x $^{(m)}$ }   x$^{(i)}$ ε $\mathbb{R}$

- Fit model p(x) by setting
  - $\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$
  - $\Sigma = \frac{1}{m} \sum_{i=1}^{m}(x^{(i)} - \mu)(x^{(i)} - \mu)^{T}$

- Given a new example x, compute
  - p(x) = $\frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}}$ exp (- $\frac{1}{2}(x-\mu)^{T}\Sigma^{-1}(x-\mu)$)
  - Flag an anomaly if p(x) < ε



Original Model

- $p(x_{1;}\ \mu_{1;}\ \sigma^2{}_1) \times p(x_{2;}\ \mu_{2;}\ \sigma^2{}_2) \times \ldots\ p(x_{n;}\ \mu_{n;}\ \sigma^2{}_n)$
- Manually create features to capture anomalies where x_1, x_2 take unusual combinations of values
- Computationally cheaper (alternatively, scales better to large n)
- Okay even if m (training set size) is small

Multivariate Gaussian

- $p(x;\ \mu\ ;\ \Sigma) =$

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

- Automatically captures correlations between features
- Computationally more expensive
- Must have m > n, or else Σ is non invertible