

# Week 1

## Clustering

[K-means Intuition](#)

[K-means algorithm](#)

[K-means Formula](#)

[Initializing K-means](#)

[Choosing the number of clusters \(k\)](#)

## Anomaly Detection

[Anomaly Detection Use Cases](#)

[Density Estimation](#)

[Density Estimation Formula](#)

[Gaussian Distribution](#)

[Parameter Estimation](#)

[Anomaly Detection Algorithm](#)

[Anomaly Detection Algorithm Example](#)

[Developing and Evaluating Anomaly Detection System](#)

[Anomaly Detection vs Supervised Learning](#)

[More Use Cases](#)

[Choosing Features for Anomaly Detection](#)

[Error Analysis](#)

# Clustering

## K-means Intuition

Randomly initialize centroids

1. Assign each points to its closest centroid
2. Recompute the avg location of the points and move the centroid
3. Repeat from 1.

## K-means algorithm

- Edge case: What if a cluster has no points?
  - Eliminate the cluster (more common)
  - Reinitialize the cluster centroid

## K-means algorithm

Randomly initialize  $K$  cluster centroids

Repeat {

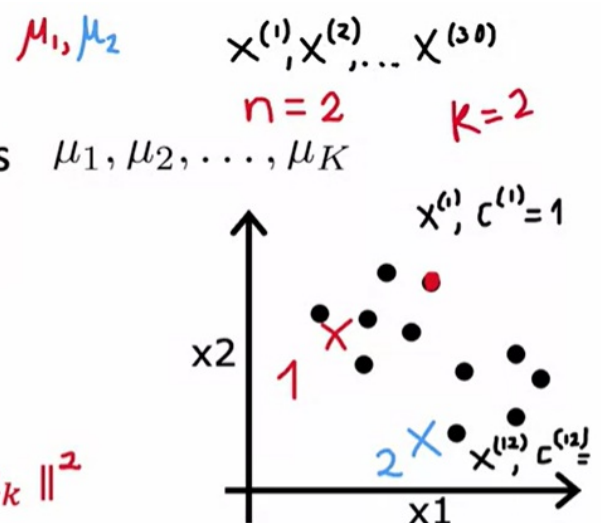
*# Assign points to cluster centroids*

for  $i = 1$  to  $m$

$c^{(i)} :=$  index (from 1 to  $K$ ) of cluster

centroid closest to  $x^{(i)}$

$\min_k \|x^{(i)} - \mu_k\|^2$



# Move cluster centroids

for  $k = 1$  to  $K$

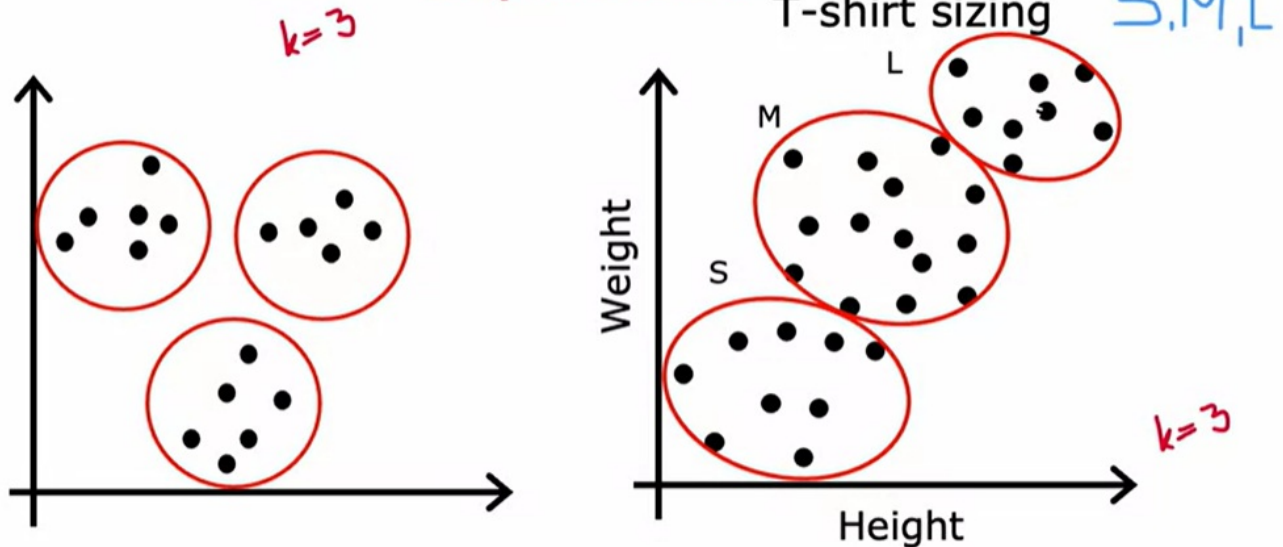
$\mu_k :=$  average (mean) of points assigned to cluster  $k$

}

$$\mu_1 = \frac{1}{4} [x^{(1)} + x^{(5)} + x^{(6)} + x^{(10)}]$$

- K-means for clusters that are not well separated:

## K-means for clusters that are not well separated



### K-means Formula

### K-means optimization objective

$c^{(i)}$  = index of cluster  $(1, 2, \dots, K)$  to which example  $x^{(i)}$  is currently assigned

$\mu_k$  = cluster centroid  $k$

$\mu_{c^{(i)}}$  = cluster centroid of cluster to which example  $x^{(i)}$  has been assigned

### Cost function

$$J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K) = \frac{1}{m} \sum_{i=1}^m \|x^{(i)} - \mu_{c^{(i)}}\|^2$$

$\min_{c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K} J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K)$

Distortion

- K-means will converge → if cost increase, there is something wrong
- If it remains the same, can stop running
- If it becomes slow after many iterations, you can stop as well

## Initializing K-means

- Run K-means multiple times to find the best initialization
- Find one with the smallest J (Cost)

## Random initialization

For  $i = 1$  to 100 { 50-1000

Randomly initialize K-means. ← k random examples

Run K-means. Get  $c^{(1)}, \dots, c^{(m)}, \mu_1, \mu_1, \dots, \mu_k$  ←

Computer cost function (distortion)

$J(c^{(1)}, \dots, c^{(m)}, \mu_1, \mu_1, \dots, \mu_k)$  ←

}

Pick set of clusters that gave lowest cost J

## Choosing the number of clusters (k)

- Elbow method
  - Idea: Look at the point where the **rate** of decrease on cost is the fastest
  - Don't choose k just to minimize the cost function!!!

## Anomaly Detection

- Problem e.g: Installing airplane engine.
  - Could have an anomaly with high temperature and low vibration of engine
  - Needs to be addressed

## Anomaly Detection Use Cases

## Anomaly detection example

how often log in?  
how many web pages visited?  
transactions?  
posts? typing speed?

Fraud detection:

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

perform additional checks to identify real fraud vs. false alarms

Manufacturing:

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

airplane engine  
circuit board  
smartphone

ratios

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.

## Density Estimation

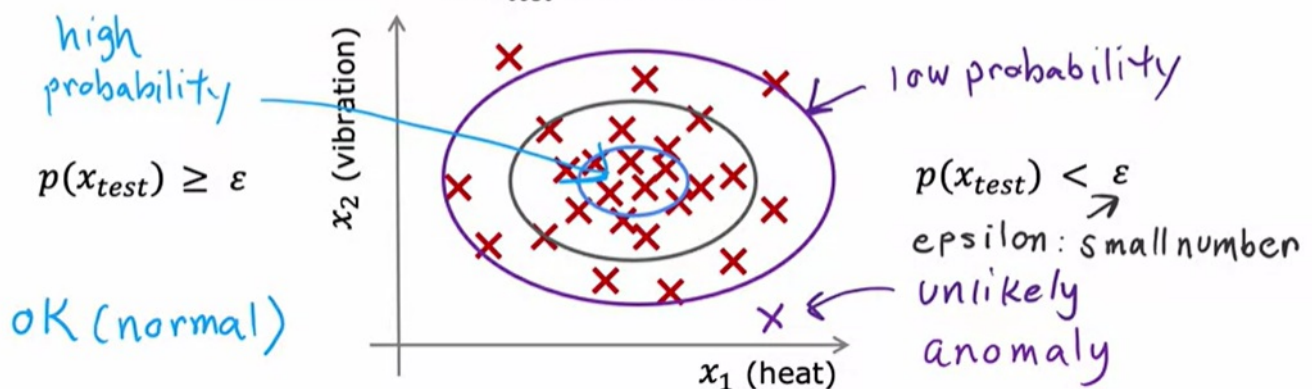
- Probability of  $x$  being seen in the dataset
- Region with high probability and low probability
- If  $p(x_{\text{test}}) < \epsilon$ , data will be flagged as an anomaly

## Density estimation

Dataset:  $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$  probability of  $x$  being seen in dataset

Model  $p(x)$

Is  $x_{\text{test}}$  anomalous?



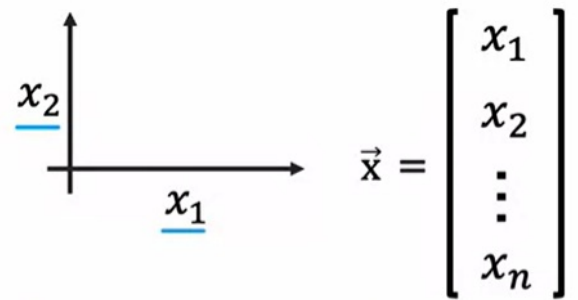
## Density Estimation Formula

- Recall that  $x$  = heat,  $y$  = vibration



# Density estimation

Training set:  $\{\vec{x}^{(1)}, \vec{x}^{(2)}, \dots, \vec{x}^{(m)}\}$   
 Each example  $\vec{x}^{(i)}$  has  $n$  features



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

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

$\sum$  "add"  
 $\Pi$  "multiply"

$p(x_1 = \text{high temp}) = 1/10$   
 $p(x_2 = \text{high vibra}) = 1/20$   
 $p(x_1, x_2) = p(x_1) * p(x_2)$   
 $= \frac{1}{10} * \frac{1}{20} = \frac{1}{200}$

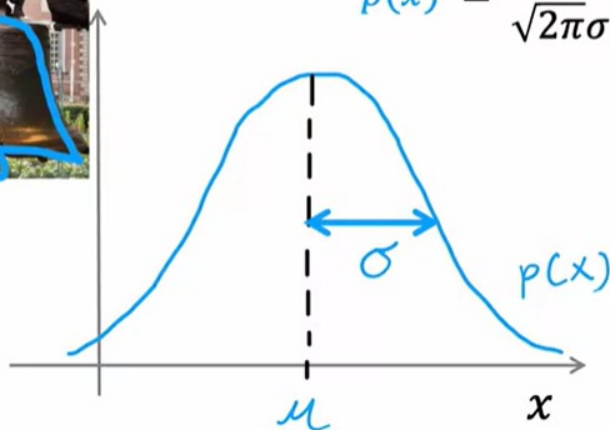
## Gaussian Distribution

### Gaussian (Normal) distribution

$\sigma$  standard deviation  
 $\sigma^2$  variance

Say  $x$  is a number.

Probability of  $x$  is determined by a Gaussian with mean  $\mu$ , variance  $\sigma^2$ .



$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$\pi = 3.14$

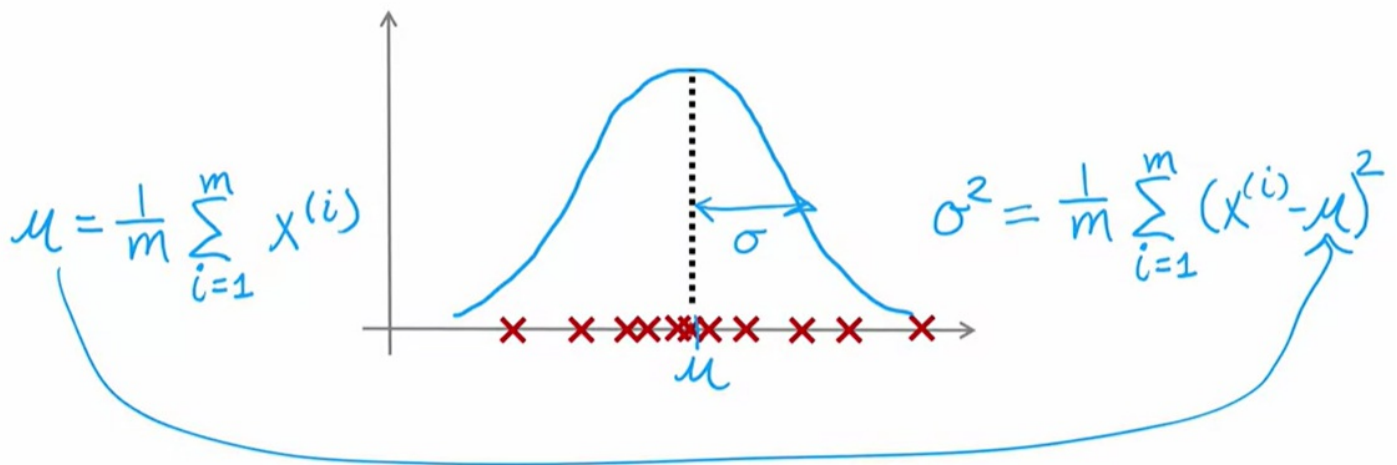


### Parameter Estimation

- Maximum likelihood estimates formula
- Note:  $1/m - 1/m - 1$  for variance

# Parameter estimation

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



## Anomaly Detection Algorithm

- Attempt to form a bell-shaped curve on the current datasets. Any data outside of the curve will have a low probability, and thus flagged as an anomaly

## Anomaly detection algorithm

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

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

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

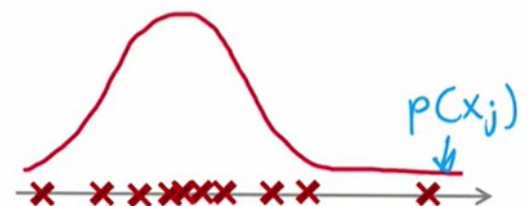
Vectorized formula

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

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

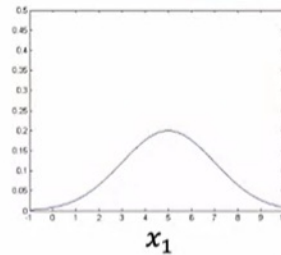
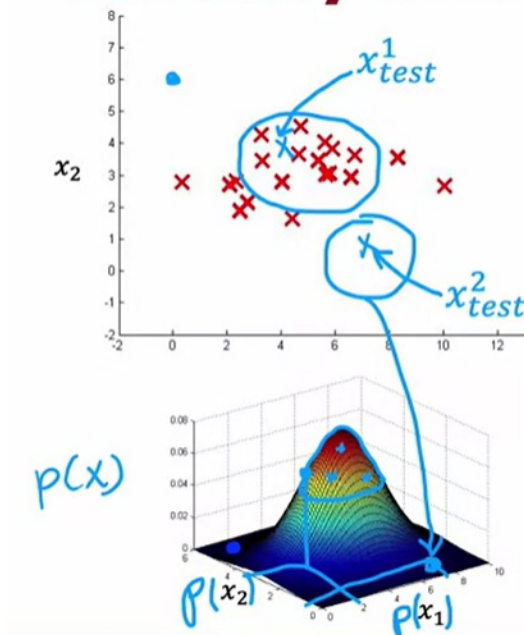
$$p(x) = \prod_{j=1}^n p(x_j; \mu_j, \sigma_j^2) = \prod_{j=1}^n \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}\right)$$

Anomaly if  $p(x) < \varepsilon$



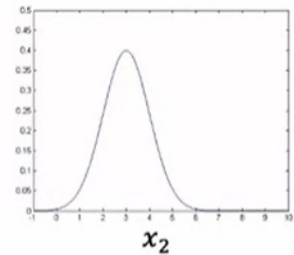
## Anomaly Detection Algorithm Example

# Anomaly detection example



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

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



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

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

$$\varepsilon = 0.02$$

$$p(x_{test}^{(1)}) = \underline{0.0426} \longrightarrow \text{"ok"}$$

$$p(x_{test}^{(2)}) = \underline{0.0021} \longrightarrow \text{anomaly}$$

## Developing and Evaluating Anomaly Detection System

- Training set, cross validation set, test set
- Test set can be omitted if very few labeled anomalous examples
- Caveat: Higher risk of overfitting, can't test the model in the future

## Aircraft engines monitoring example

10000 good (normal) engines  
~~20~~ 20 flawed engines (anomalous)  
~~2~~ 2

2 to 50  
 $y=1$

Training set: 6000 good engines

train algorithm on training set

CV: 2000 good engines ( $y=0$ )  
 use cross validation set

10 anomalous ( $y=1$ )

tune  $\varepsilon$  tune  $x_j$

Test: 2000 good engines ( $y=0$ ),

10 anomalous ( $y=1$ )

Alternative: No test set Use if very few labeled anomalous examples

Training set: 6000 good engines 2

CV: 4000 good engines ( $y=0$ ), ~~20~~ 20 anomalous ( $y=1$ )

tune  $\varepsilon$  tune  $x_j$

- Predicting an anomaly:



# Algorithm evaluation

course 2 week 3  
skewed datasets

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

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

10  
2000

Possible evaluation metrics:

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

Use cross validation set to choose parameter  $\varepsilon$

## Anomaly Detection vs Supervised Learning

- How to decide between supervised learning and anomaly detection
- More variations → use anomaly detection
- Less types of data → supervised learning

### Anomaly detection vs. Supervised learning

Very small number of positive examples ( $y = 1$ ). (0-20 is common).  
Large number of negative examples ( $y = 0$ ).  
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 nothing like any of the anomalous examples we've seen so far.

Fraud

Large number of positive and negative examples.

20 positive 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

More Use Cases



## Anomaly detection

- Fraud detection
- Manufacturing - Finding new previously unseen defects in manufacturing. (e.g. aircraft engines)
- Monitoring machines in a data center
- ⋮

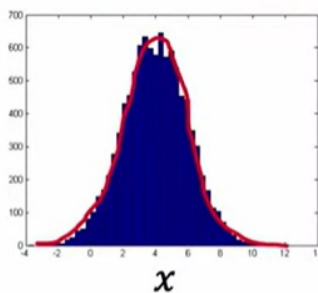
## vs. Supervised learning

- Email spam classification
- Manufacturing - Finding known, previously seen defects  $y=1$   
scratches
- Weather prediction (sunny/rainy/etc.)
- Diseases classification
- ⋮

### Choosing Features for Anomaly Detection

- Select features that makes your data look Gaussian

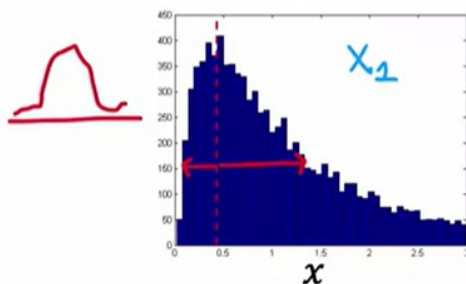
## Non-gaussian features



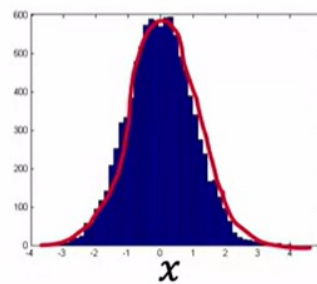
$$p(x_1; \mu_1, \sigma_1^2)$$

`plt.hist(x)`

$$\begin{aligned} x_1 &\leftarrow \log(x_1) \\ x_2 &\leftarrow \log(x_2 + 1) \quad \log(x_2 + c) \\ x_3 &\leftarrow \sqrt{x_3} = x_3^{1/2} \\ x_4 &\leftarrow x_4^{1/3} \end{aligned}$$



`np.log(x)`



### Error Analysis

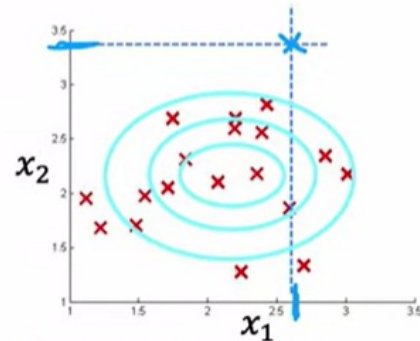
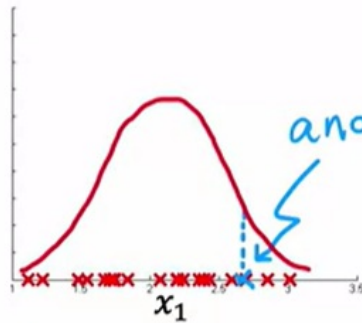
- If  $p(x)p(x)p(x)$  for both normal and anomalous is large, anomalous data can be missed
- Choose features that are likely to be more variable

# Error analysis for anomaly detection

Want  $p(x) \geq \epsilon$  large for normal examples  $x$ .  
 $p(x) < \epsilon$  small for anomalous examples  $x$ .

Most common problem:

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



$x_1$  num transactions

$x_2$  typing speed