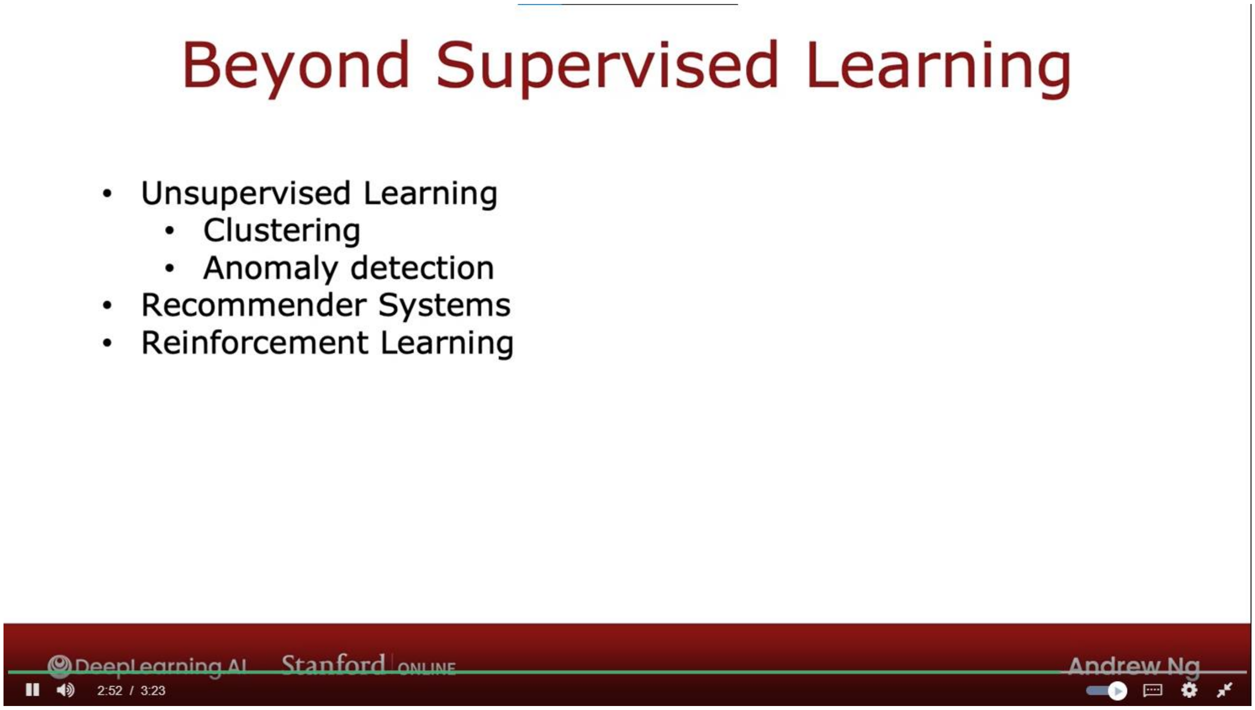
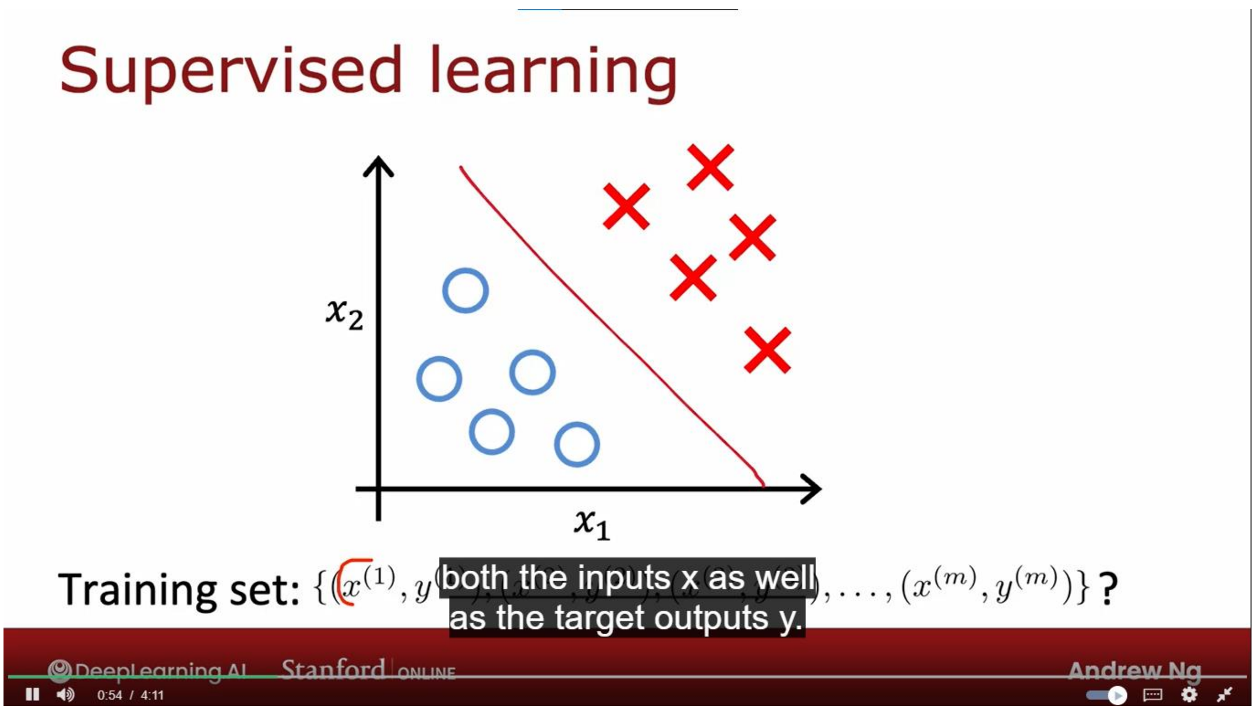
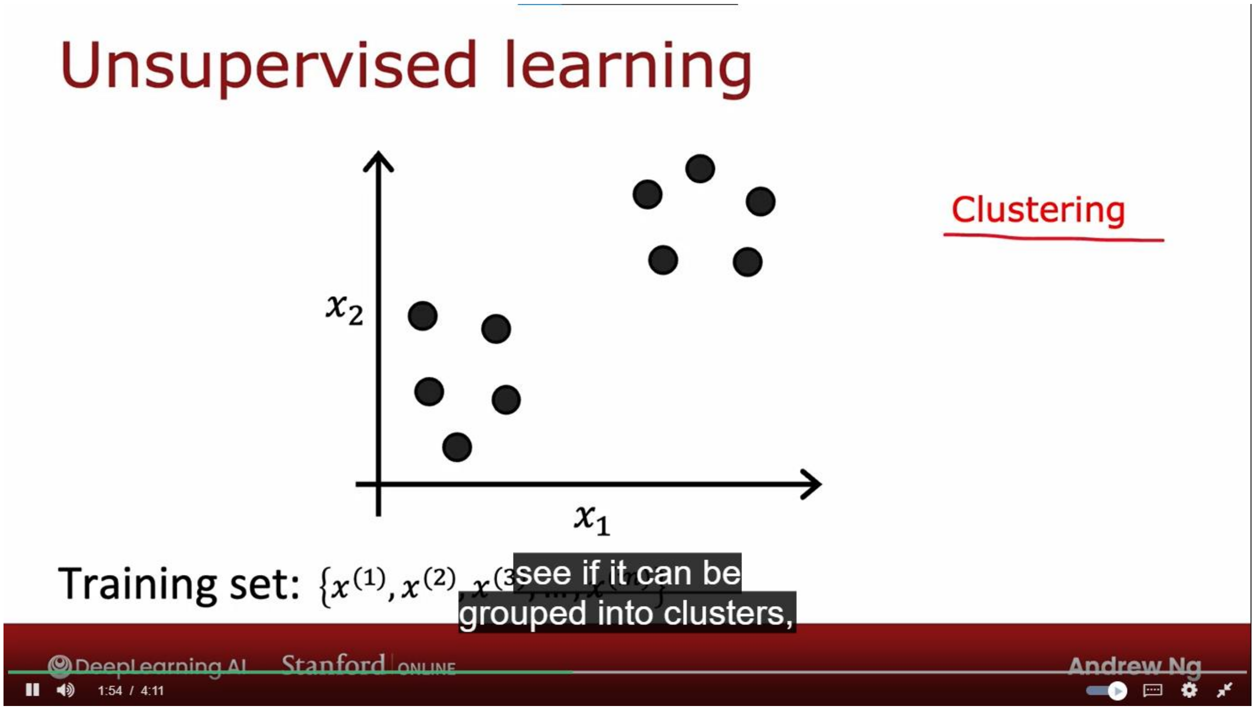
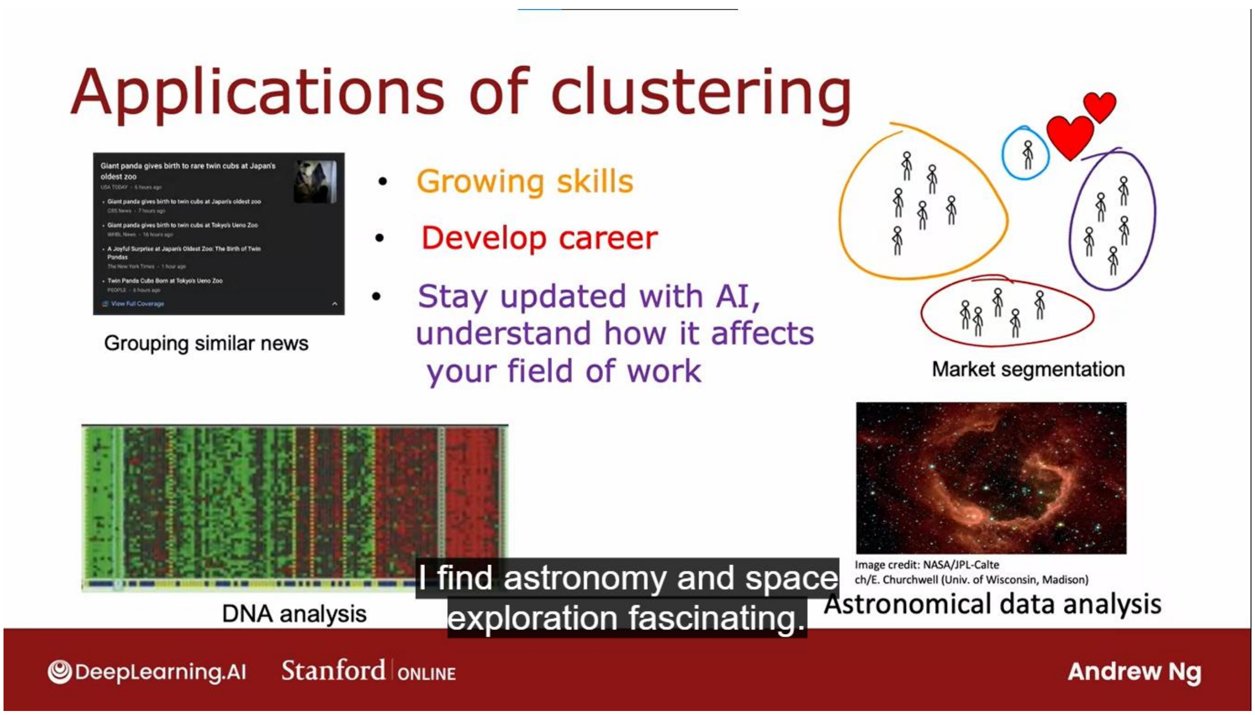
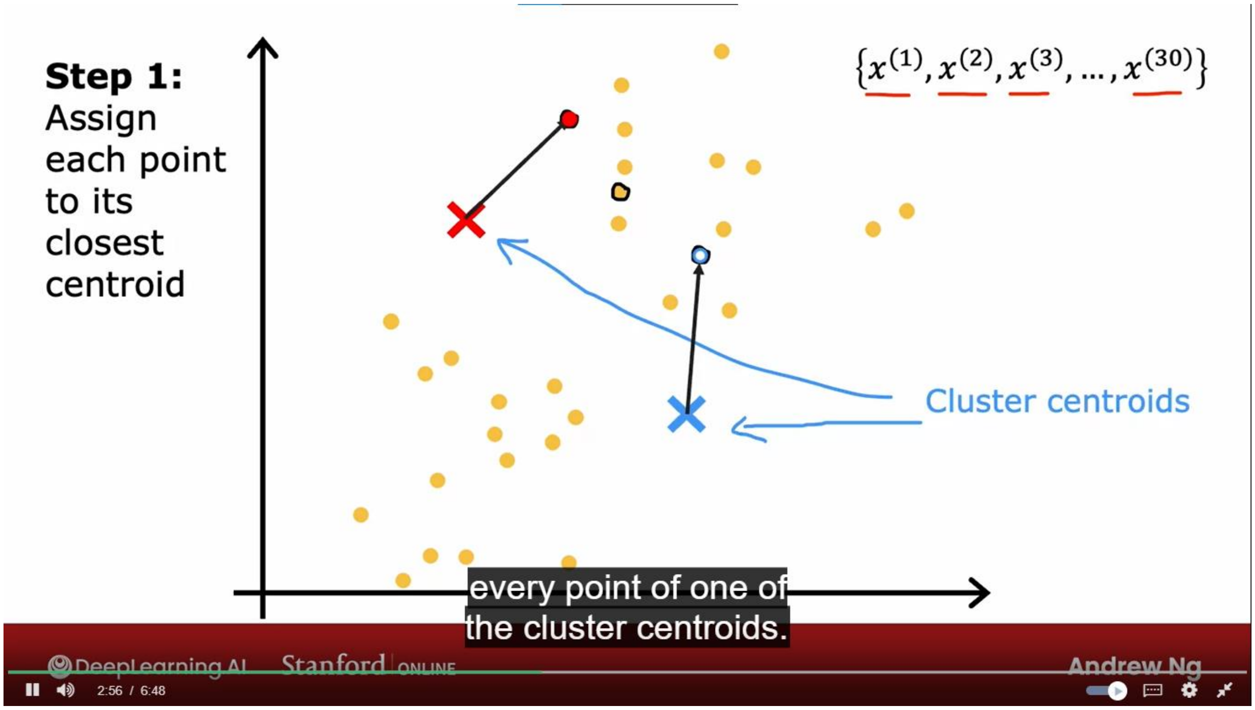
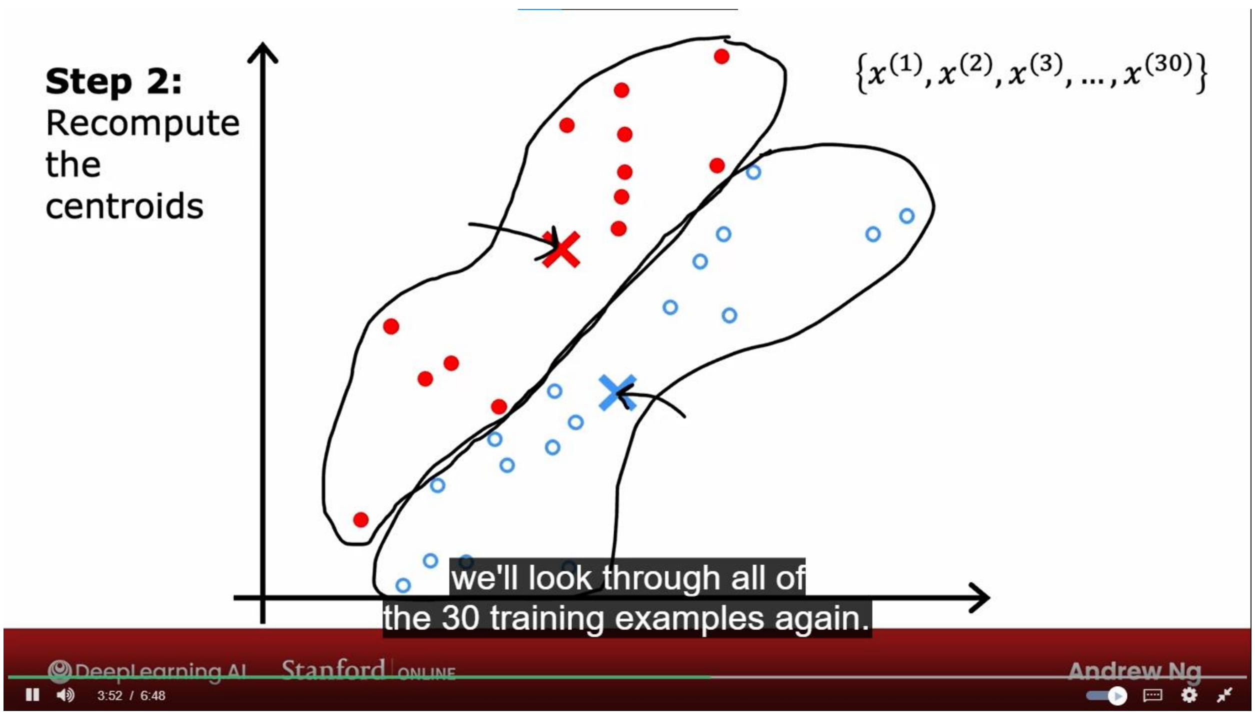

Beyond Supervised Learning
• Unsupervised Learning
• Clustering
• Anomaly detection
• Recommender Systems
• Reinforcement Learning
DeepLearning.AI
Stanford ONLINE
Andrew Ng
2:52 / 3:23

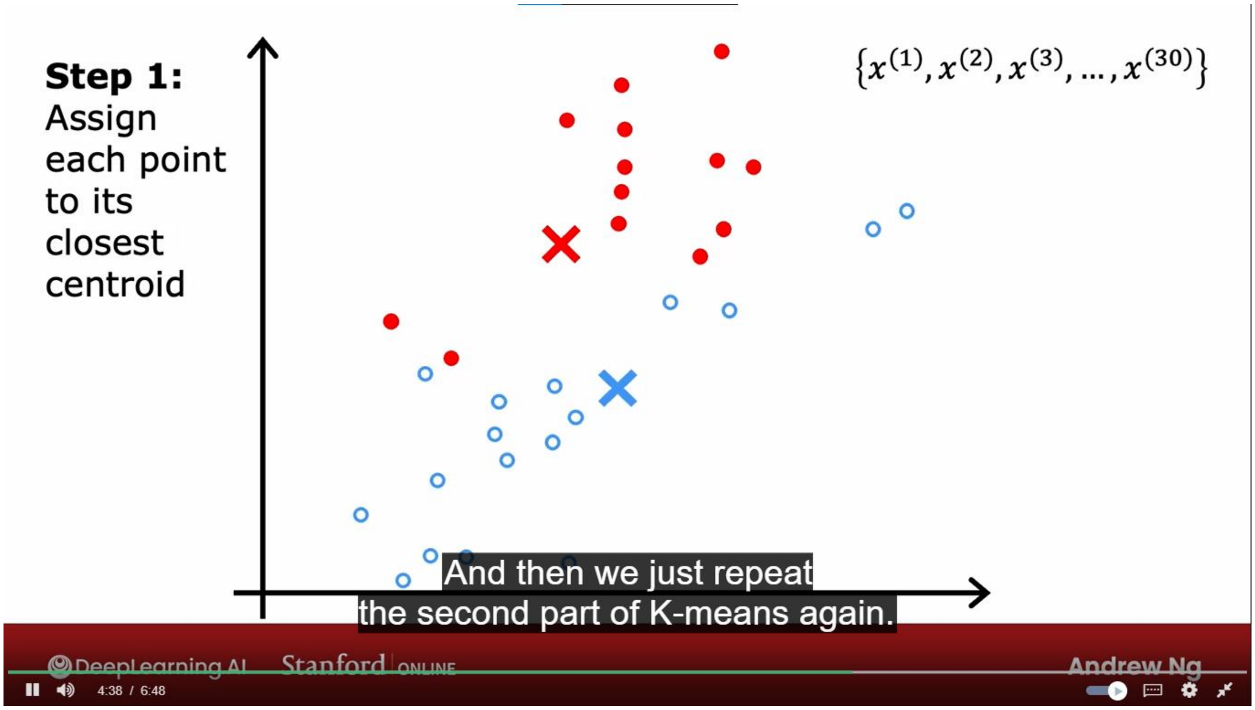

Supervised learning
x_2
x_1
Training set: $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$?
both the inputs x as well
as the target outputs y.
DeepLearning.AI
Stanford ONLINE
Andrew Ng
0:54 / 4:11

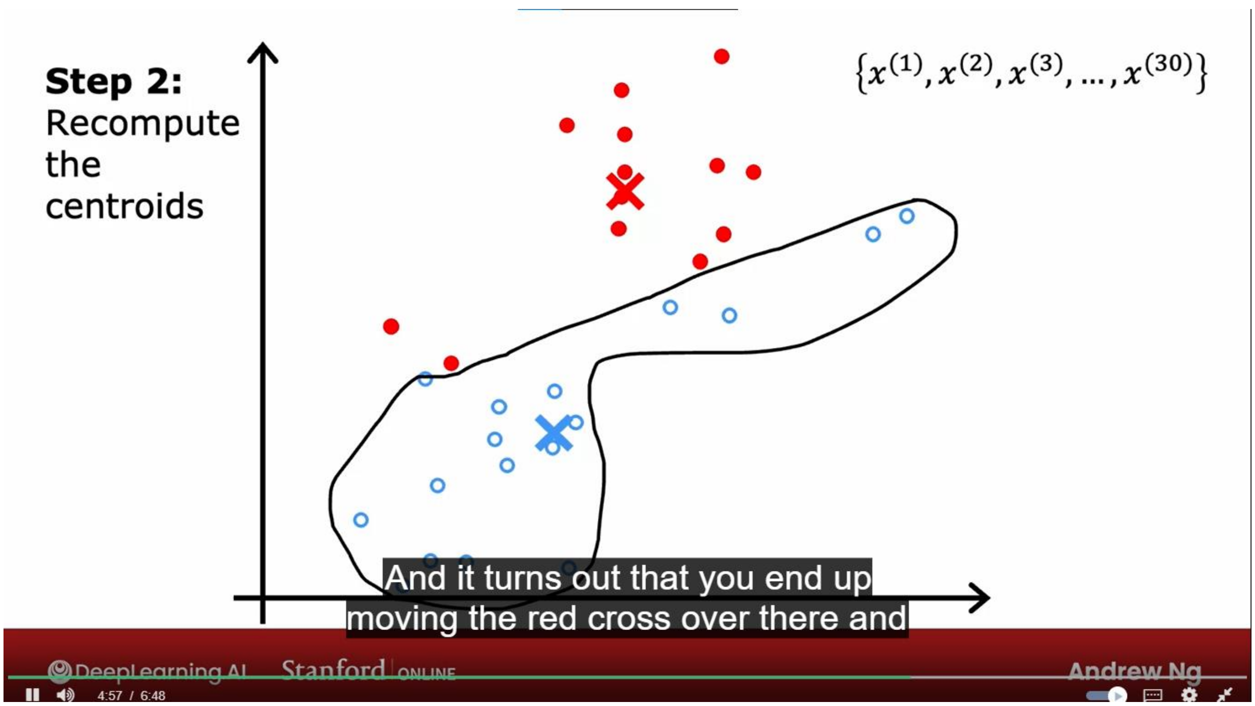

Unsupervised learning
Clustering
x_2
x_1
Training set: $\{x^{(1)}, x^{(2)}, x^{(3}$
see if it can be
grouped into clusters,
DeepLearning.AI
Stanford ONLINE
Andrew Ng
1:54 / 4:11

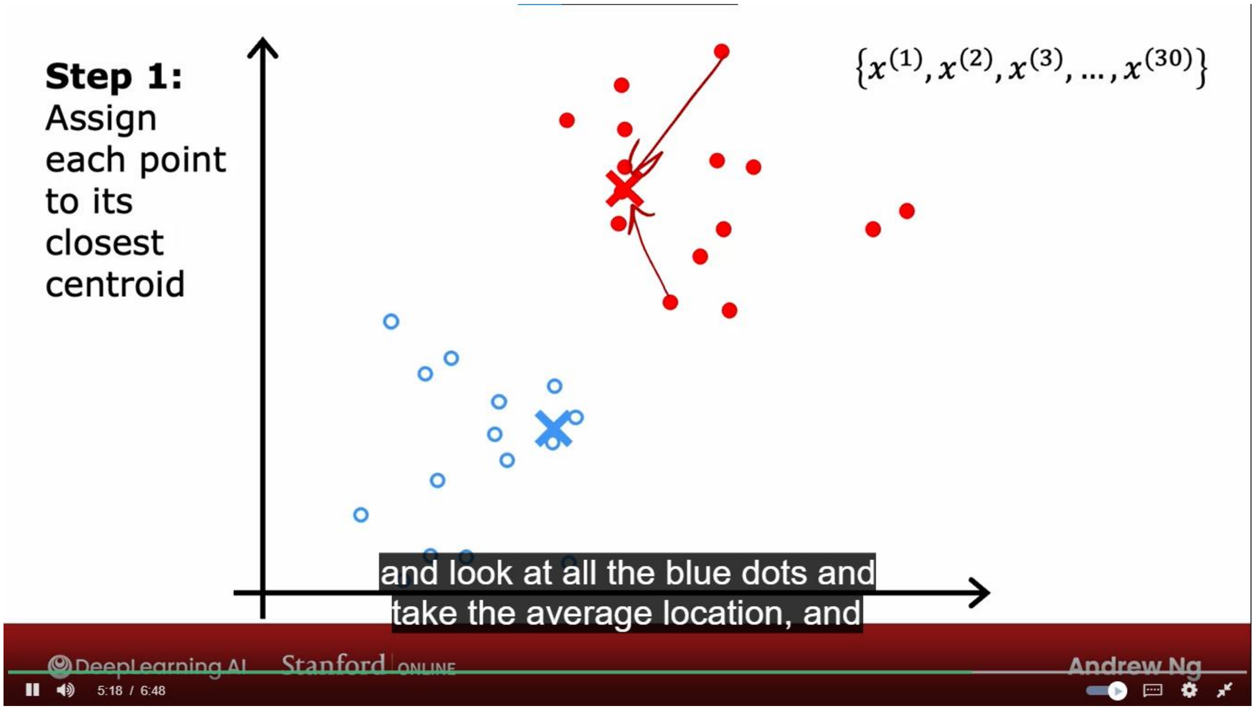

Applications of clustering
Growing skills
Develop career
Stay updated with AI, understand how it affects your field of work
Grouping similar news
Market segmentation
DNA analysis
Image credit: NASA/JPL-Caltech/E. Churchwell (Univ. of Wisconsin, Madison)
Astronomical data analysis
I find astronomy and space
exploration fascinating.
DeepLearning.AI
Stanford ONLINE
Andrew Ng

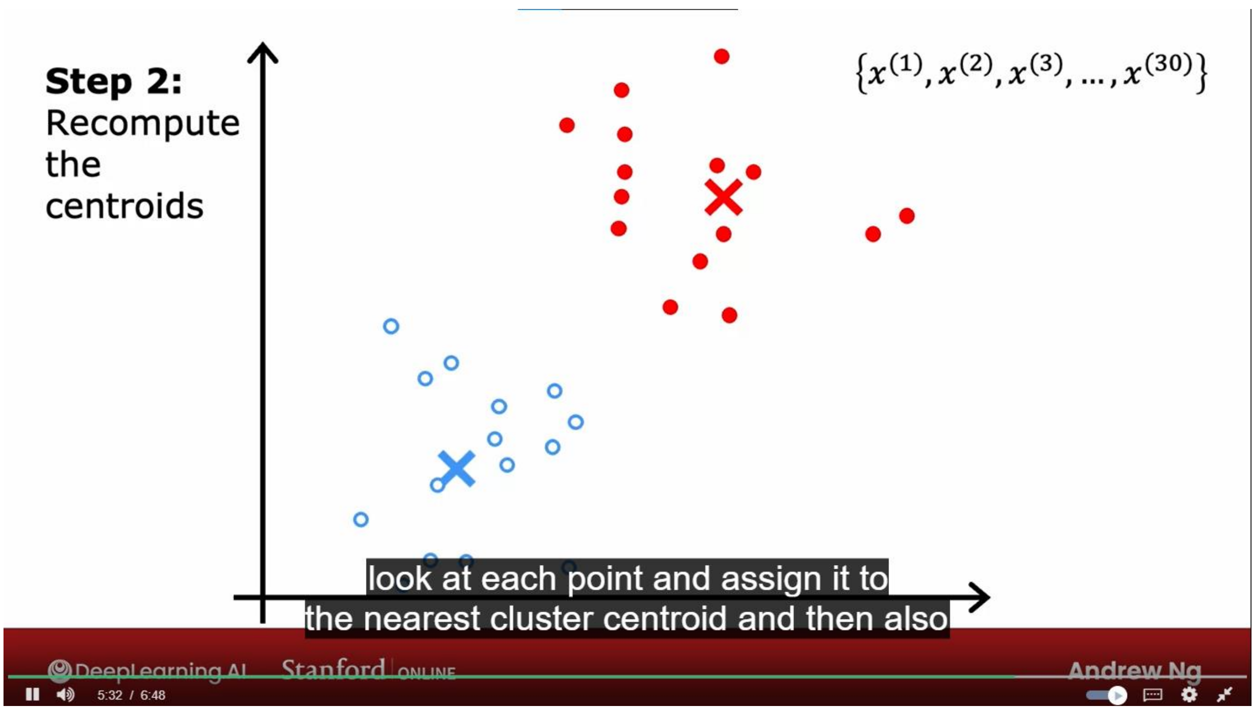

Step 1:
Assign each point to its closest centroid
$\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(30)}\}$
Cluster centroids
every point of one of the cluster centroids.
DeepLearning.AI
Stanford ONLINE
Andrew Ng
2:56 / 6:48

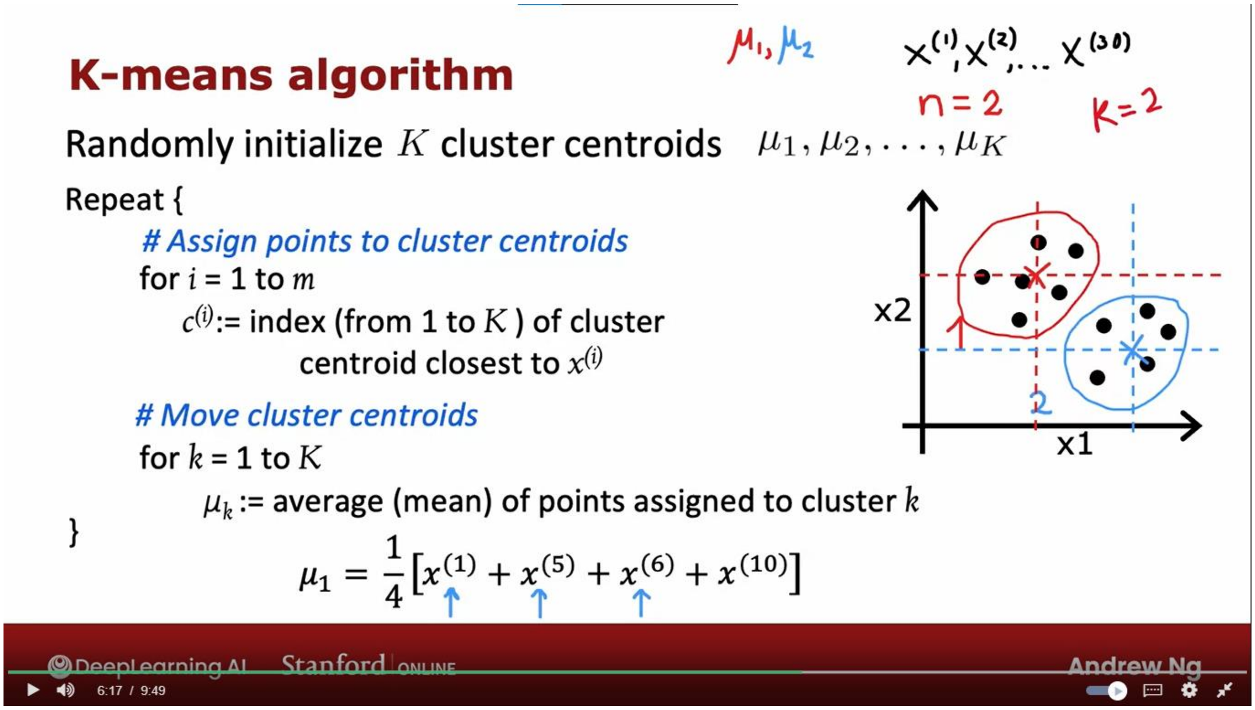

Step 2:
Recompute the centroids
$\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(30)}\}$
we'll look through all of the 30 training examples again.
DeepLearning.AI
Stanford ONLINE
Andrew Ng
3:52 / 6:48

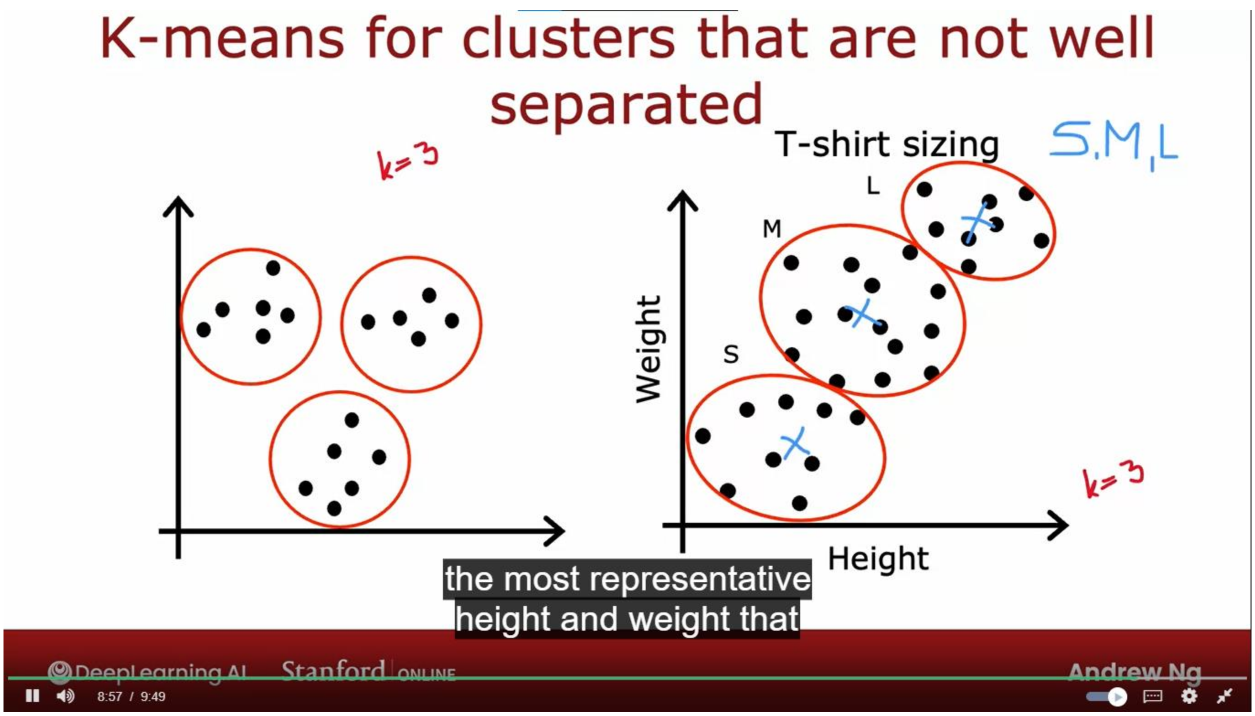

Step 1:
Assign each point to its closest centroid
{x(1), x(2), x(3), ..., x(30)}
And then we just repeat
the second part of K-means again.
DeepLearning.AI
Stanford ONLINE
Andrew Ng
4:38 / 6:48


Step 2:
Recompute the centroids
{x(1), x(2), x(3), ..., x(30)}
And it turns out that you end up
moving the red cross over there and
DeepLearning.AI
Stanford ONLINE
Andrew Ng
4:57 / 6:48


Step 1:
Assign each point to its closest centroid
$\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(30)}\}$
and look at all the blue dots and
take the average location, and
DeepLearning.AI
Stanford ONLINE
Andrew Ng
5:18 / 6:48


Step 2:
Recompute the centroids
$\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(30)}\}$
look at each point and assign it to
the nearest cluster centroid and then also
DeepLearning.AI
Stanford ONLINE
Andrew Ng
5:32 / 6:48


K-means algorithm
μ_1, μ_2
$x^{(1)}, x^{(2)}, \ldots x^{(30)}$
n = 2
K = 2
Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K$
Repeat {
# Assign points to cluster centroids
for i = 1 to m
$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$
Move cluster centroids
for k = 1 to K
μ_k := average (mean) of points assigned to cluster k
}
$\mu_1 = \frac{1}{4}\left[x^{(1)} + x^{(5)} + x^{(6)} + x^{(10)}\right]$
x2
x1
1
2
DeepLearning.AI
Stanford ONLINE
Andrew Ng
6:17 / 9:49


K-means for clusters that are not well separated
k=3
T-shirt sizing
S, M, L
L
M
S
Weight
Height
k=3
the most representative
height and weight that
DeepLearning.AI
Stanford ONLINE
Andrew Ng
8:57 / 9:49

K-means optimization objective

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

μ_k = cluster centroid k

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

$x^{(10)}$ $c^{(10)}$ $\mu_{c^{(10)}}$

Cost function

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

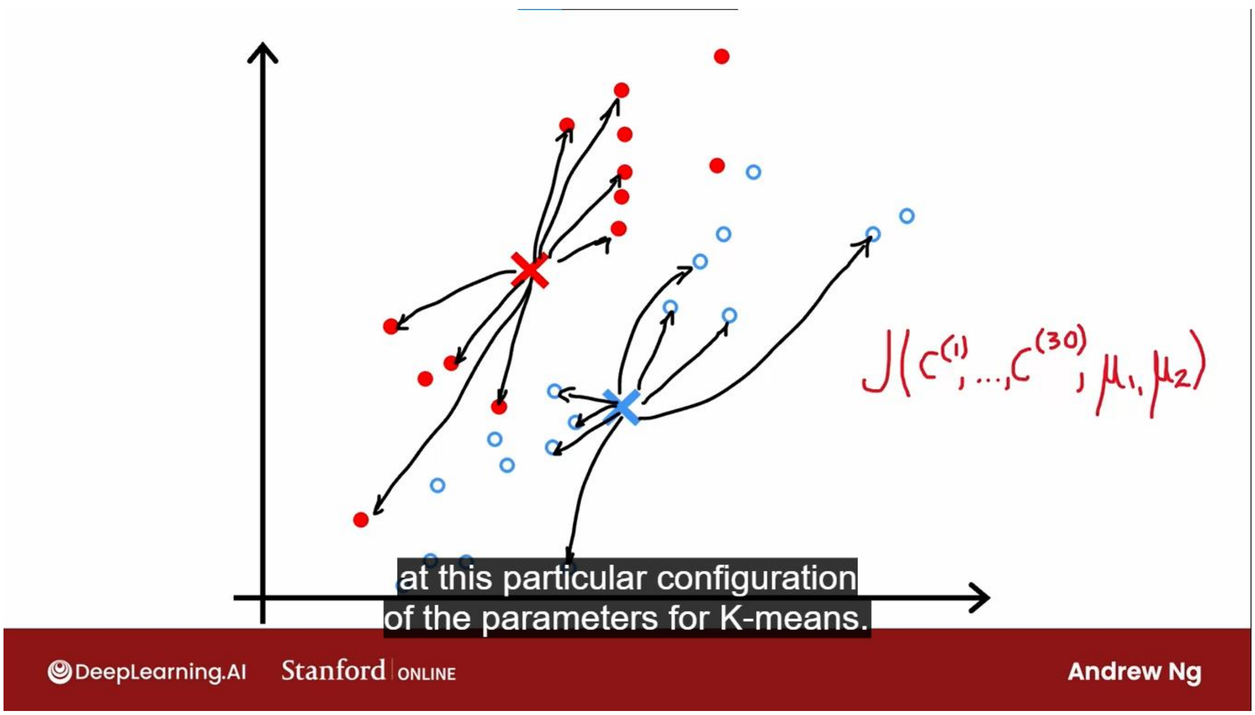
$$\min_{\substack{c^{(1)}, \ldots, c^{(m)} \\ \mu_1, \ldots, \mu_K}} J\left(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K\right)$$

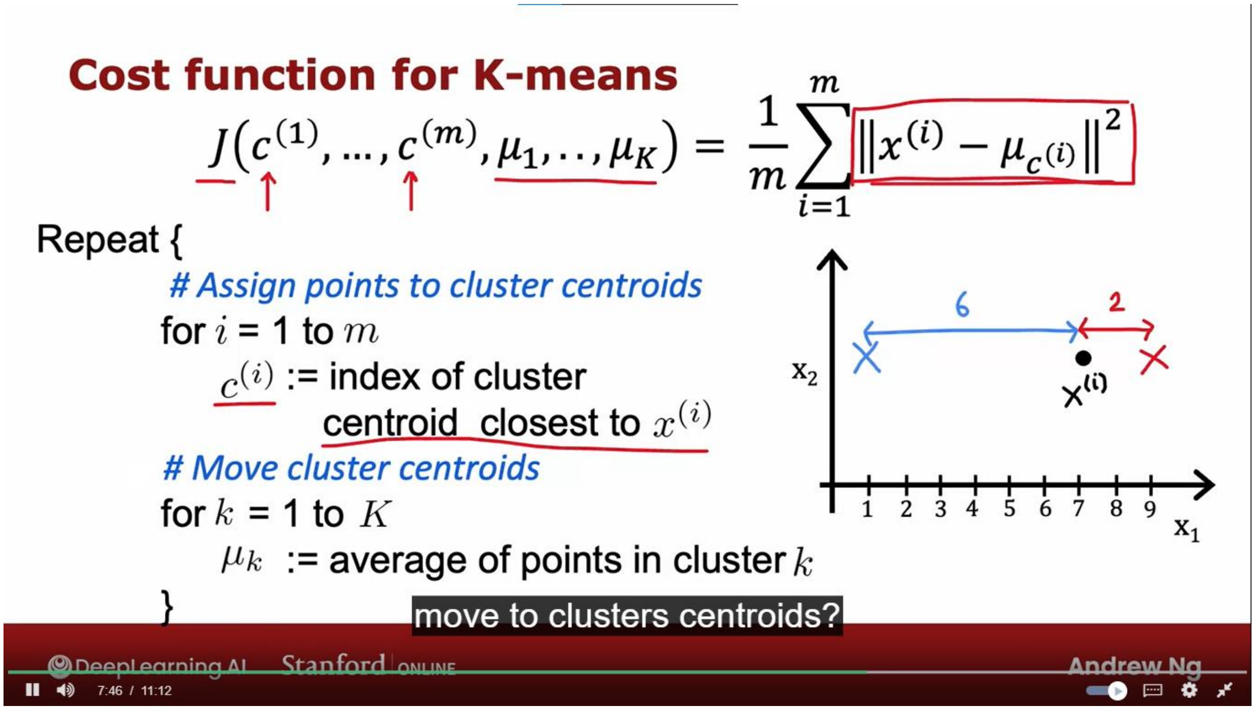
Distortion

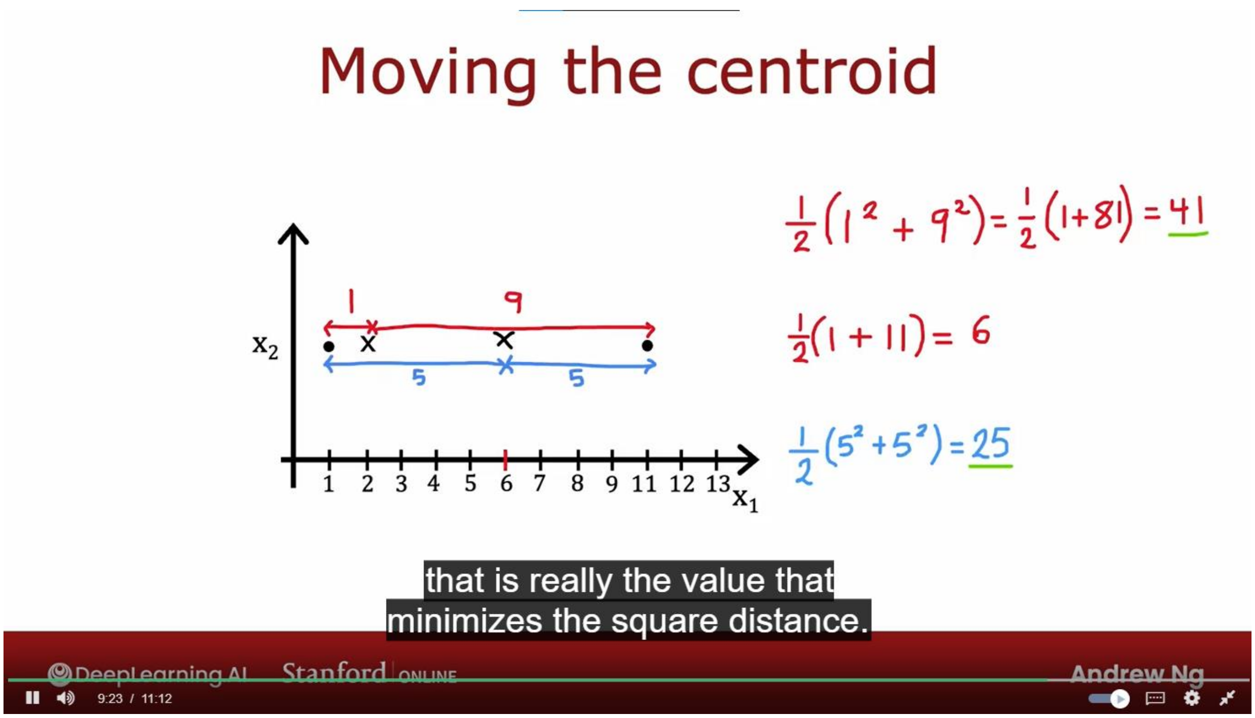
the distortion cost function, that's just what this formula J is computing.

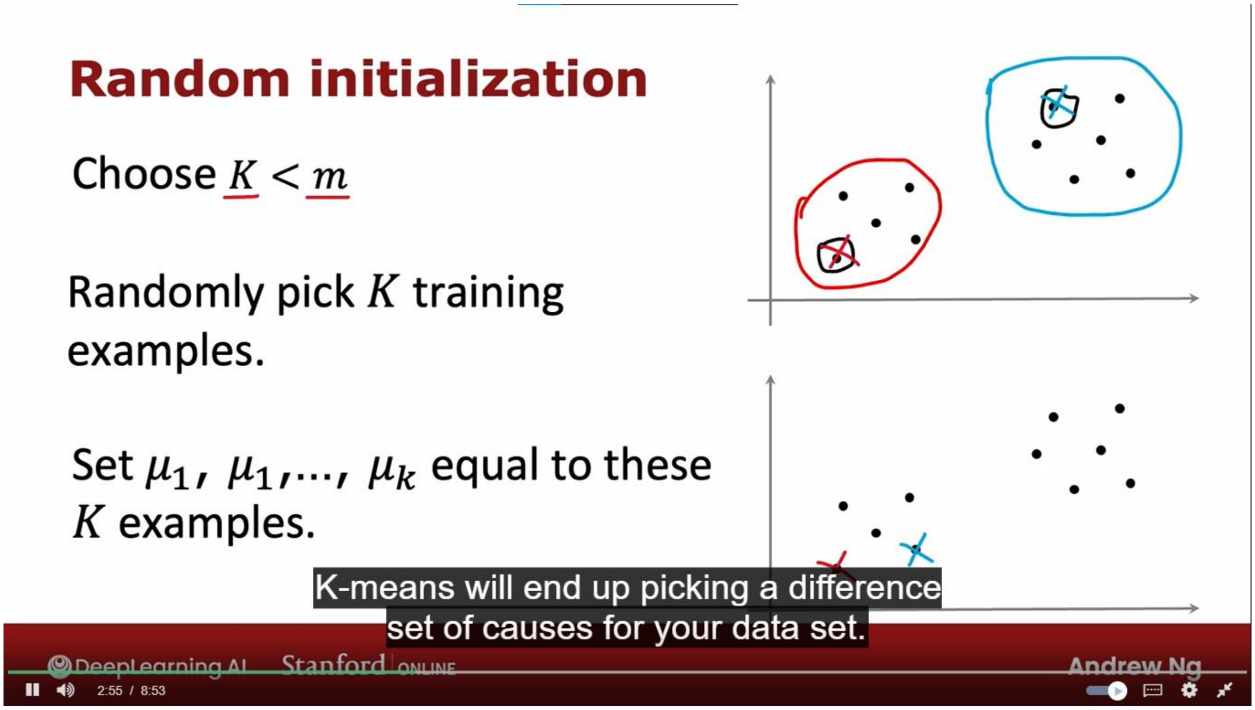
DeepLearning.AI Stanford ONLINE Andrew Ng

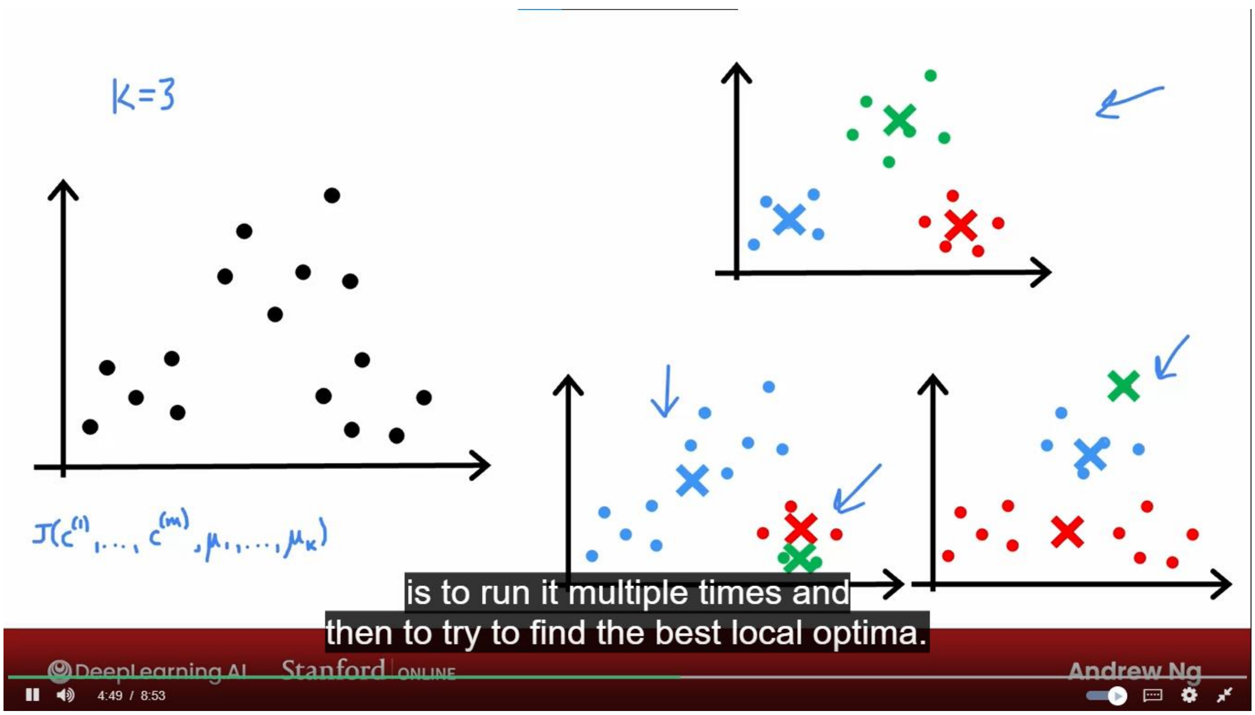
5:04 / 11:12




Cost function for K-means
$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K) = \frac{1}{m}\sum_{i=1}^{m} \boxed{\left\|x^{(i)} - \mu_{c^{(i)}}\right\|^2}$
Repeat {
# Assign points to cluster centroids
for i = 1 to m
$c^{(i)}$:= index of cluster centroid closest to $x^{(i)}$
Move cluster centroids
for k = 1 to K
μ_k := average of points in cluster k
}
6
2
x_2
$x^{(i)}$
1 2 3 4 5 6 7 8 9
x_1
move to clusters centroids?
DeepLearning.AI
Stanford ONLINE
Andrew Ng
7:46 / 11:12


Moving the centroid
$\frac{1}{2}(1^2 + 9^2) = \frac{1}{2}(1+81) = 41$
$\frac{1}{2}(1 + 11) = 6$
$\frac{1}{2}(5^2 + 5^2) = 25$
1
9
5
5
x_2
1 2 3 4 5 6 7 8 9 11 12 13
x_1
that is really the value that
minimizes the square distance.
DeepLearning.AI
Stanford ONLINE
Andrew Ng
9:23 / 11:12


Random initialization
Choose $K < m$
Randomly pick K training examples.
Set μ_1, μ_1,..., μ_k equal to these K examples.
K-means will end up picking a difference set of causes for your data set.
DeepLearning.AI Stanford ONLINE
Andrew Ng
2:55 / 8:53


K=3
$J(c^{(1)},\ldots,c^{(m)},\mu_1,\ldots,\mu_K)$
is to run it multiple times and then to try to find the best local optima.
DeepLearning.AI Stanford ONLINE
Andrew Ng
4:49 / 8:53

Random initialization

For $i = 1$ to 100 {

Randomly initialize K-means. ← k random examples

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

Computer cost function (distortion)

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

}

Pick set of clusters that gave lowest cost J

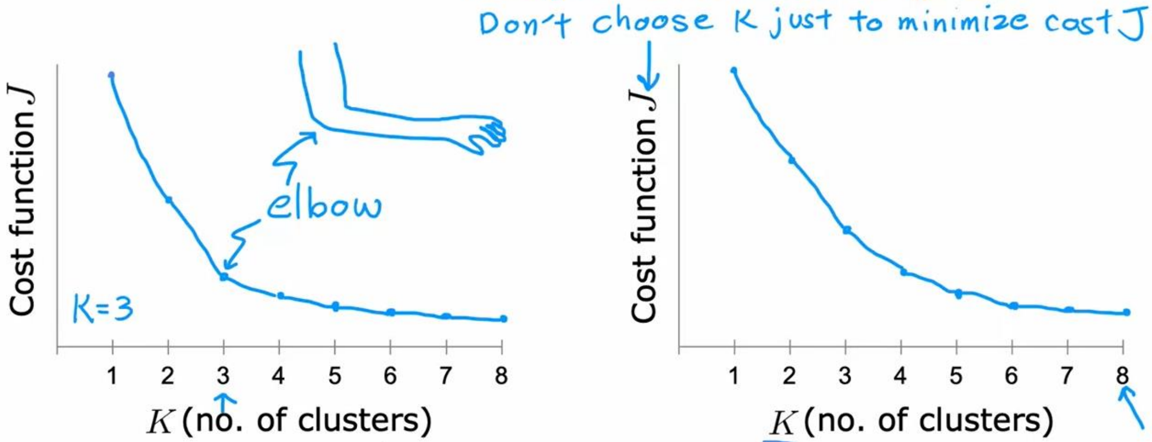
I plugged in the number up here as 100.

Choosing the value of K

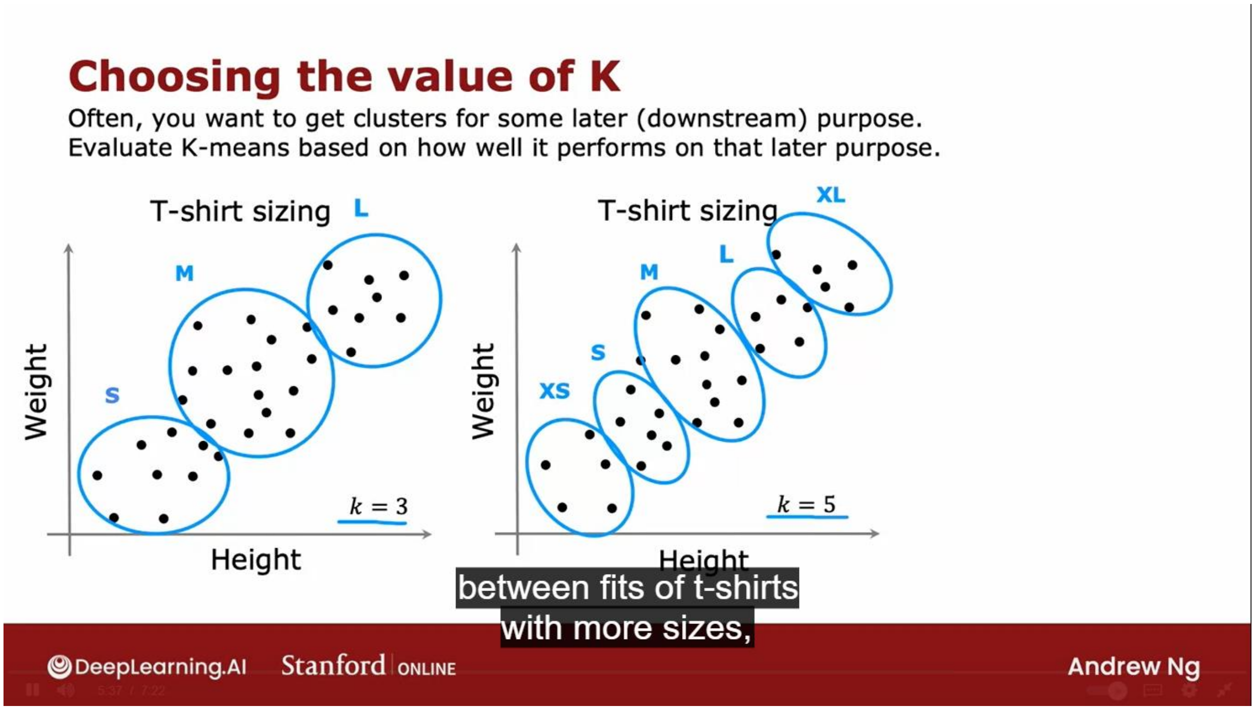
Elbow method

the right "K" is often ambiguous

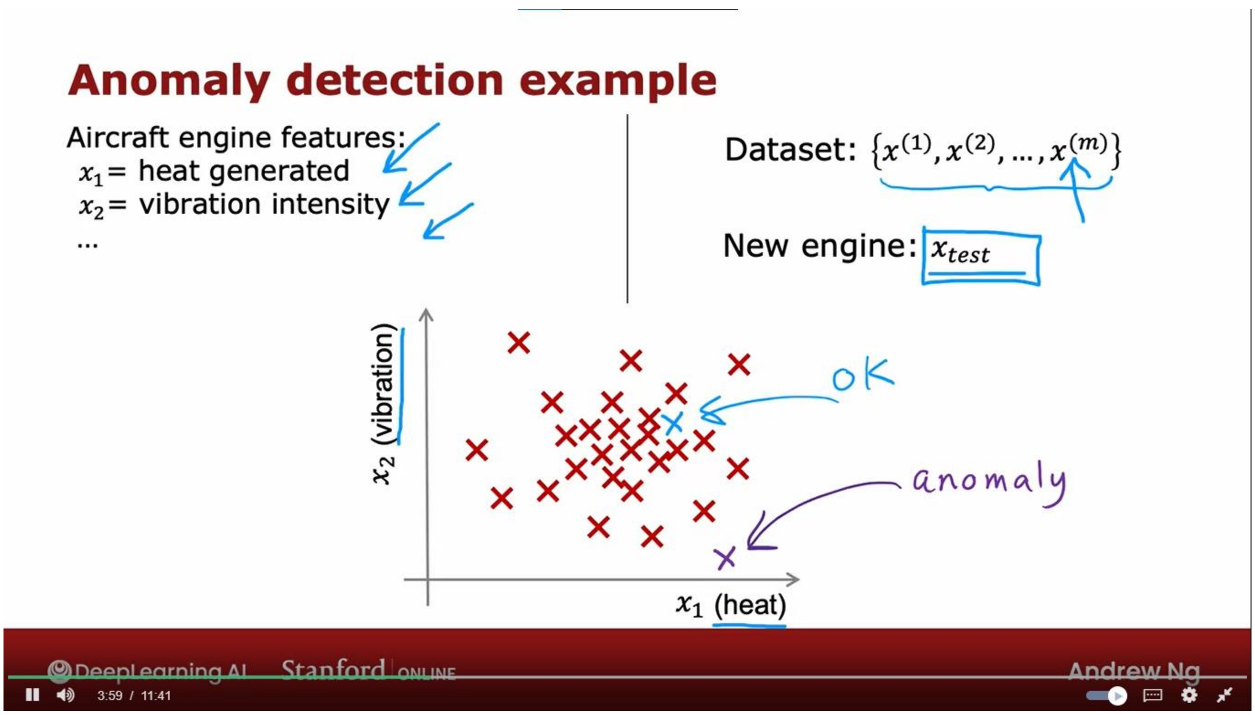
Don't choose K just to minimize cost J



Choosing K to minimize



Choosing the value of K
Often, you want to get clusters for some later (downstream) purpose.
Evaluate K-means based on how well it performs on that later purpose.
T-shirt sizing
S
M
L
Weight
Height
$k = 3$
T-shirt sizing
XS
S
M
L
XL
Weight
Height
$k = 5$
between fits of t-shirts
with more sizes,
DeepLearning.AI
Stanford ONLINE
Andrew Ng



Anomaly detection example
Aircraft engine features:
x_1 = heat generated
x_2 = vibration intensity
...
Dataset: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$
New engine: $\boxed{x_{test}}$
x_2 (vibration)
x_1 (heat)
ok
anomaly
DeepLearning.AI
Stanford ONLINE
Andrew Ng
3:59 / 11:41

Density estimation

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

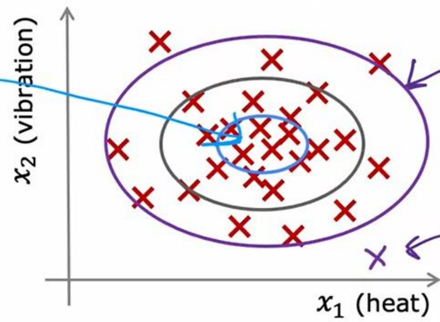
Model $p(x)$

Is x_{test} anomalous?

high probability

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

OK (normal)



low probability

$p(x_{test}) < \varepsilon$

epsilon: small number

unlikely anomaly

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 $p(x) < \varepsilon$

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.

Gaussian (Normal) distribution

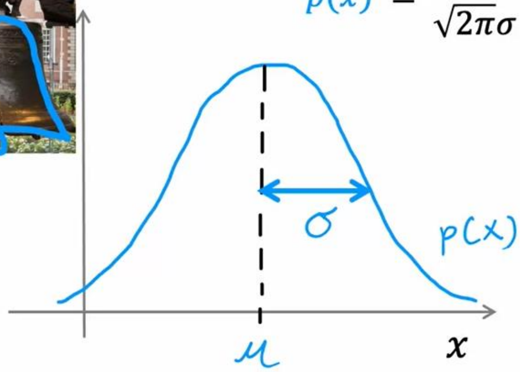
σ standard deviation

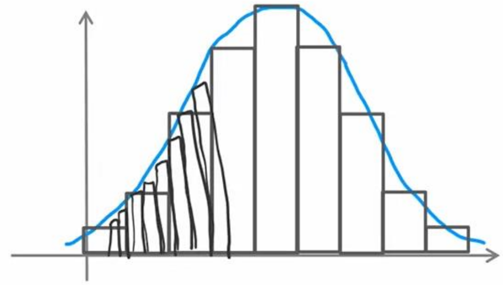
σ^2 variance

Say x is a number.
Probability of x is determined by a Gaussian with mean μ, variance σ^2.

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

$\pi = 3.14$



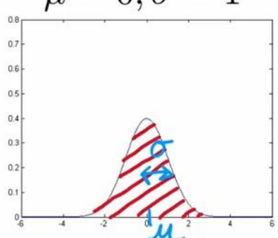


DeepLearning.AI Stanford ONLINE Andrew Ng

3:32 / 10:39

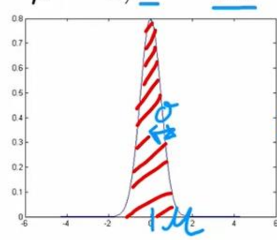
Gaussian distribution example

$\mu = 0, \sigma = 1$

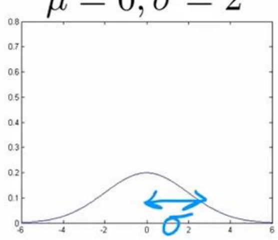


$\mu = 0, \sigma = 0.5$

$\sigma^2 = 0.25$

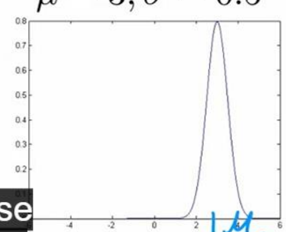


$\mu = 0, \sigma = 2$



$\sigma^2 = 4$

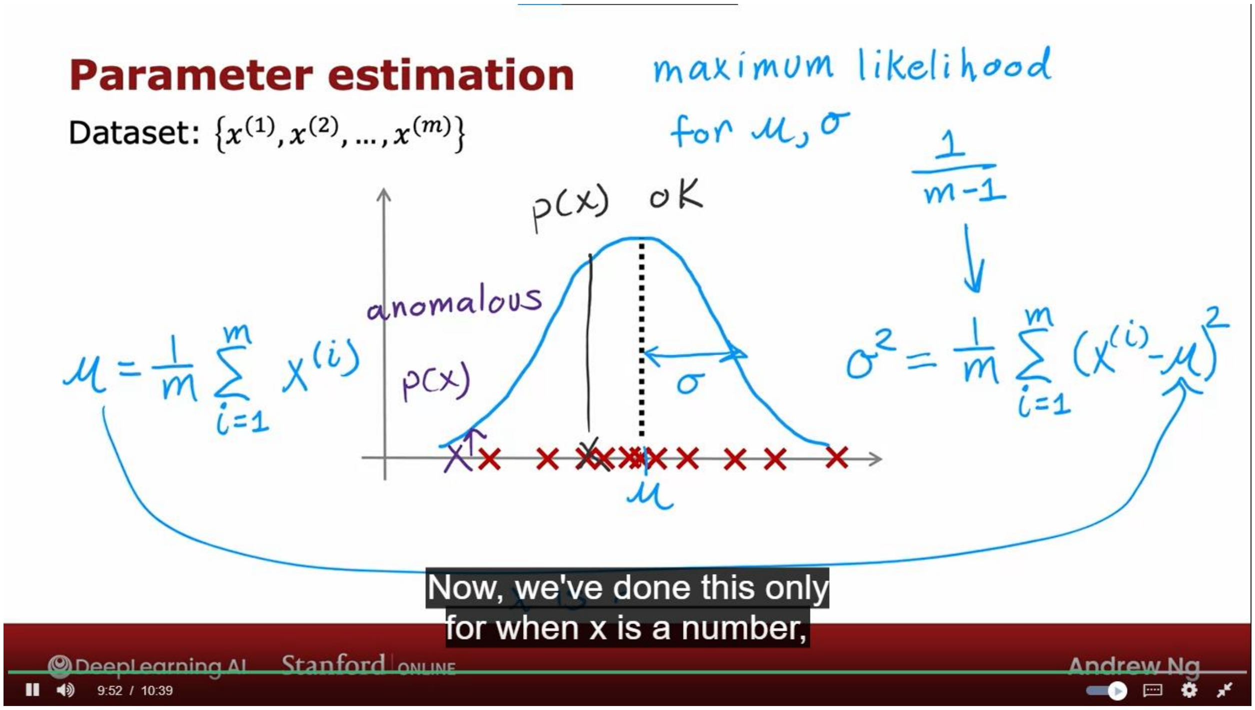
$\mu = 3, \sigma = 0.5$

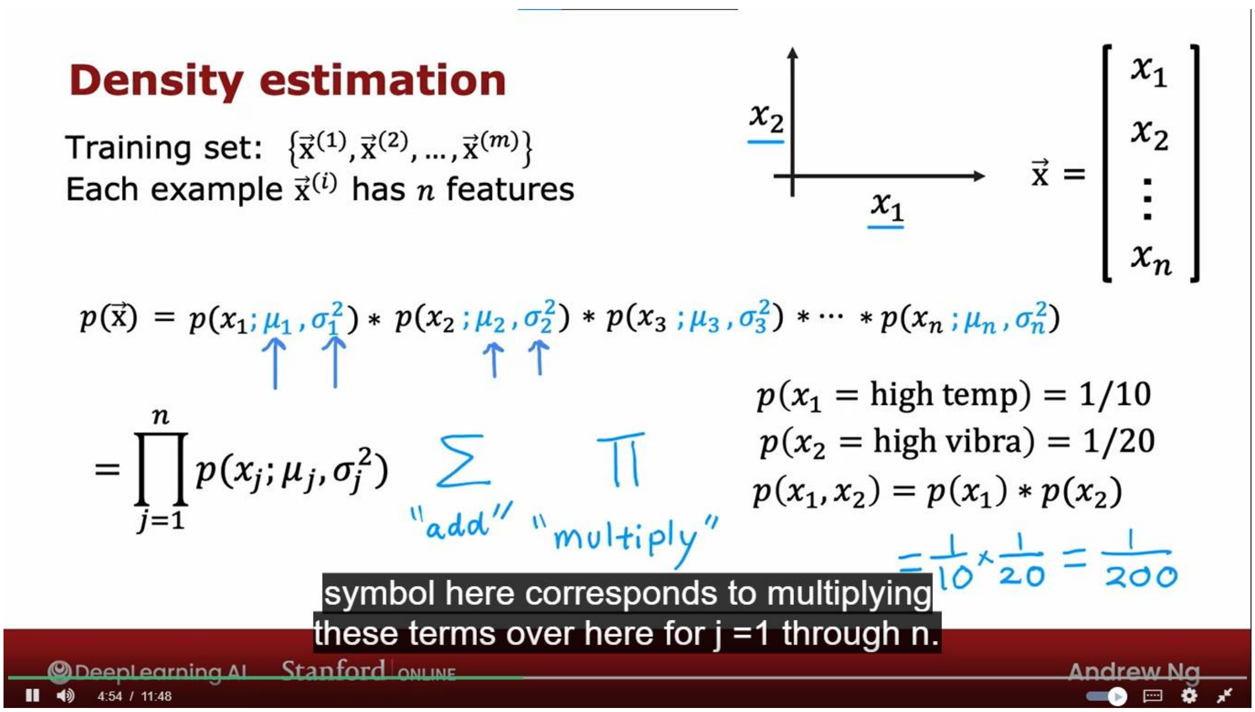


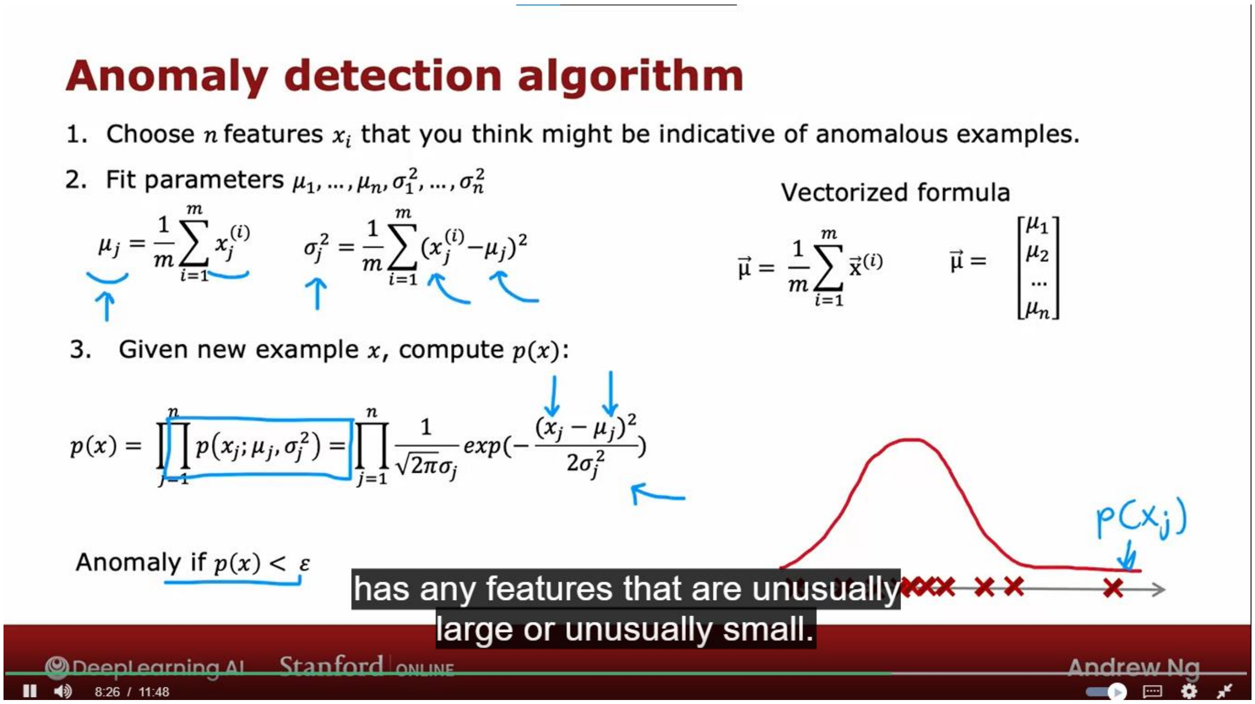
0.5 in both of these cases on the right.

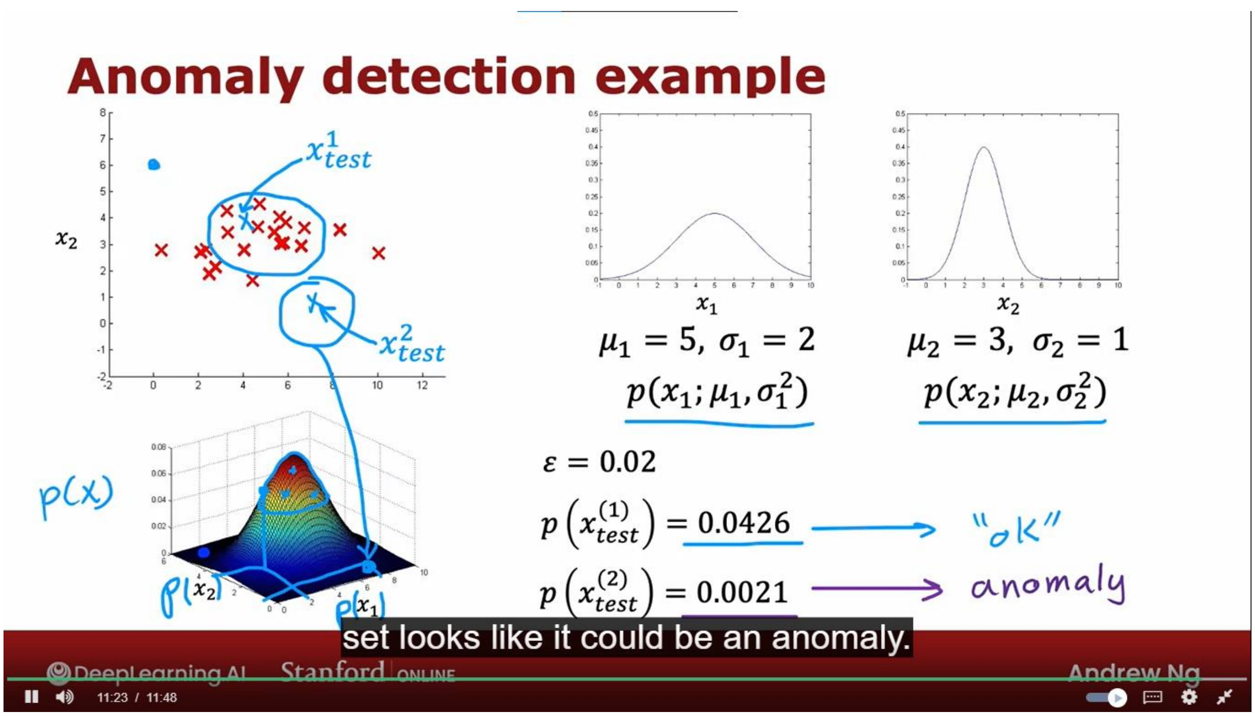
DeepLearning.AI Stanford ONLINE Andrew Ng

6:14 / 10:39


Parameter estimation
maximum likelihood for μ, σ
Dataset: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$
$\frac{1}{m-1}$
p(x) OK
anomalous p(x)
$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$
$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m} (x^{(i)} - \mu)^2$
σ
μ
Now, we've done this only
for when x is a number,
DeepLearning.AI
Stanford ONLINE
Andrew Ng
9:52 / 10:39


Density estimation
Training set: $\{\vec{x}^{(1)}, \vec{x}^{(2)}, \ldots, \vec{x}^{(m)}\}$
Each example $\vec{x}^{(i)}$ has n features
x_2
x_1
$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$
$p(\vec{x}) = p(x_1; \mu_1, \sigma_1^2) * p(x_2; \mu_2, \sigma_2^2) * p(x_3; \mu_3, \sigma_3^2) * \cdots * p(x_n; \mu_n, \sigma_n^2)$
$= \prod_{j=1}^{n} p(x_j; \mu_j, \sigma_j^2)$
$\sum$ "add"
$\prod$ "multiply"
$p(x_1 = \text{high temp}) = 1/10$
$p(x_2 = \text{high vibra}) = 1/20$
$p(x_1, x_2) = p(x_1) * p(x_2)$
$= \frac{1}{10} \times \frac{1}{20} = \frac{1}{200}$
symbol here corresponds to multiplying
these terms over here for j =1 through n.
DeepLearning.AI
Stanford ONLINE
Andrew Ng
4:54 / 11:48


Anomaly detection algorithm
1. Choose n features x_i that you think might be indicative of anomalous examples.
2. Fit parameters $\mu_1, \ldots, \mu_n, \sigma_1^2, \ldots, \sigma_n^2$
$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)}$
$\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m} (x_j^{(i)} - \mu_j)^2$
Vectorized formula
$\vec{\mu} = \frac{1}{m}\sum_{i=1}^{m} \vec{x}^{(i)}$
$\vec{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \ldots \\ \mu_n \end{bmatrix}$
3. Given new example x, compute $p(x)$:
$p(x) = \prod_{j=1}^{n} \boxed{p(x_j; \mu_j, \sigma_j^2)} = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_j} exp(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2})$
$p(x_j)$
Anomaly if $p(x) < \varepsilon$
has any features that are unusually
large or unusually small.
DeepLearning.AI
Stanford ONLINE
Andrew Ng
8:26 / 11:48


Anomaly detection example
x_2
x_{test}^1
x_{test}^2
$p(x)$
$p(x_2)$
$p(x_1)$
x_1
x_2
$\mu_1 = 5, \ \sigma_1 = 2$
$p(x_1; \mu_1, \sigma_1^2)$
$\mu_2 = 3, \ \sigma_2 = 1$
$p(x_2; \mu_2, \sigma_2^2)$
$\varepsilon = 0.02$
$p\left(x_{test}^{(1)}\right) = 0.0426 \longrightarrow$ "ok"
$p\left(x_{test}^{(2)}\right) = 0.0021 \longrightarrow$ anomaly
set looks like it could be an anomaly.
DeepLearning.AI
Stanford ONLINE
Andrew Ng
11:23 / 11:48

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 labeled data, of anomalous and non-anomalous examples.

$y=1$ $y=0$

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

$y=0$ for all training examples

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

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

include a few anomalous examples $y=1$

mostly normal examples $y=0$

but there were accidentally
labeled with y equals 0.

DeepLearning.AI Stanford ONLINE Andrew Ng

3:31 / 11:38

Aircraft engines monitoring example

10000 good (normal) engines

~~20~~ 2 flawed engines (anomalous)

2 to 50

$y=1$

$y=0$

Training set: 6000 good engines

train algorithm on training set

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

use cross validation set

tune ε tune x_j

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

Alternative: No test set

Use if very few labeled anomalous examples

higher risk of overfitting

Training set: 6000 good engines

CV: 4000 good engines ($y = 0$), ~~20~~ 2 anomalous ($y = 1$)

tune ε tune x_j

and so its performance on

DeepLearning.AI Stanford ONLINE Andrew Ng

8:14 / 11:38

Algorithm evaluation

course 2 week3
skewed datasets

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

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

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 ε

choose a good choice for
the parameter Epsilon.

DeepLearning.AI Stanford ONLINE Andrew Ng

10:58 / 11:38

Anomaly detection vs. Supervised learning

Anomaly detection	Supervised learning
Very small number of positive examples ($y = 1$). (0-20 is common). Large number of negative ($y = 0$) examples.	Large number of positive and negative examples.
p(x) y=1	20 positive examples
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.	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.
Fraud	Spam

spam because it's trying to
detect more of the types of spam

DeepLearning.AI Stanford ONLINE Andrew Ng

4:24 / 8:08

Anomaly detection vs. Supervised learning

Anomaly detection

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

⋮

Supervised learning

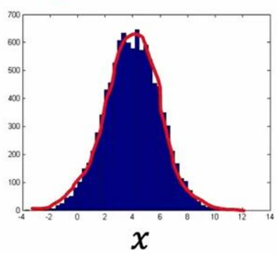
- Email spam classification
- Manufacturing - Finding known, previously seen defects $y=1$ scratches
- Weather prediction (sunny/rainy/etc.)
- Diseases classification

⋮

Then that would also tend to be a supervised learning application.

DeepLearning.AI Stanford ONLINE Andrew Ng

Non-gaussian features



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

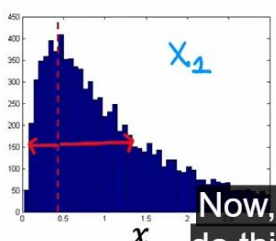
`plt.hist(x)`

$x_1 \leftarrow \log(x_1)$

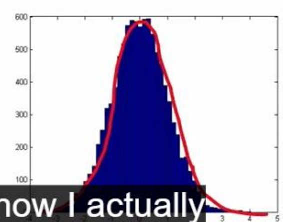
$x_2 \leftarrow \log(x_2 + 1)$ $\log(x_2 + c)$

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

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



`np.log(x)` $\longrightarrow$



Now, let me illustrate how I actually do this and that you put a notebook.

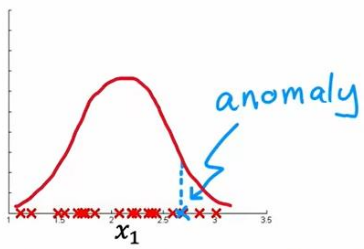
DeepLearning.AI Stanford ONLINE Andrew Ng

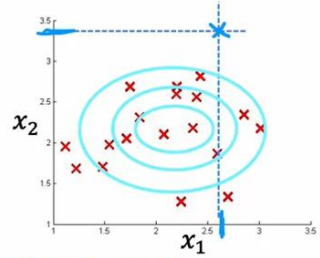
4:24 / 14:54

Error analysis for anomaly detection

Want $p(x) \geq \varepsilon$ large for normal examples x.
$p(x) < \varepsilon$ 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

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
high x_3 = CPU load
low x_4 = network traffic

not unusual

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

Deciding feature choice based on $p(x)$
Large for normal examples;
Becomes small for anomaly in the cross validation set