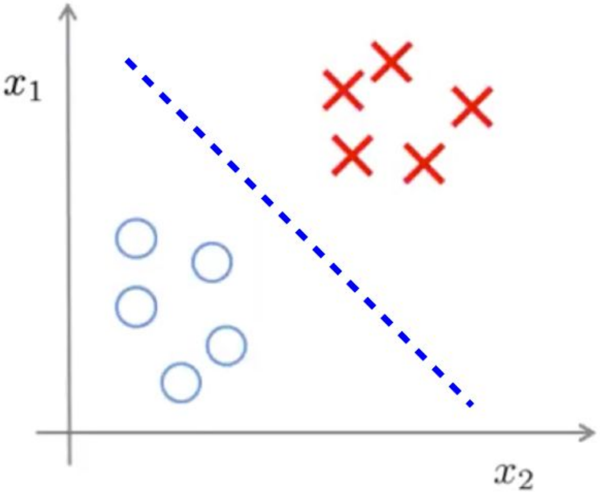
Unsupervised Learning

(การเรียนรู้แบบไม่มีผู้สอน / ไม่ใช้ข้อมูลสอน)

สำหรับทำ Clustering

Krittameth Teachasrisaksakul

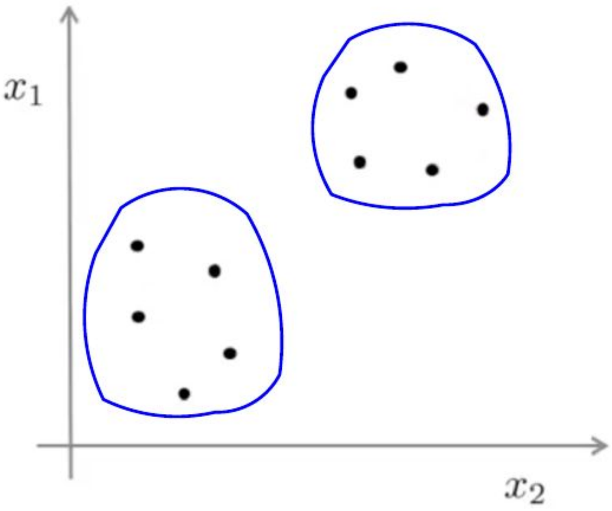
Supervised Learning (ทบทวน)



ชุดข้อมูล Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots, (x^{(m)}, y^{(m)})\}$

Figure source: Andrew Ng's Machine Learning course

Unsupervised Learning



เราอยากให้ unsupervised learning algorithm หาโครงสร้างบางอย่างในข้อมูล

ถ้าเรามีชุดข้อมูลนี้ โครงสร้างที่เราอยากให้ algorithm หา คือ:

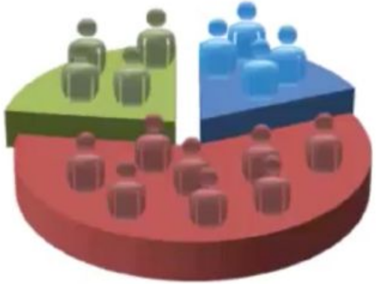
- clustering

นอกจากนี้ ยังมีโครงสร้างชนิดอื่นๆ ที่เราอาจหาได้ แต่เราจะพูดถึงมัน หลังจากนี้

ชุดข้อมูล Training set: $\{(x^{(1)}, \cancel{y^{(1)}}), (x^{(2)}, \cancel{y^{(2)}}), (x^{(3)}, y^{(3)}), \ldots, (x^{(m)}, y^{(m)})\}$

Figure source: Andrew Ng's Machine Learning course

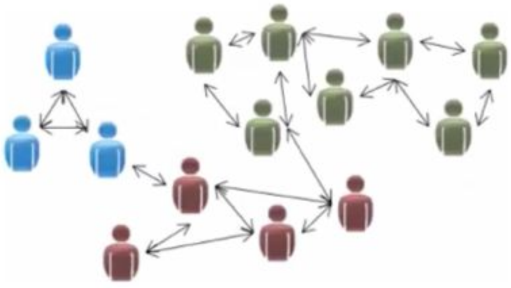
Clustering ใช้ทำอะไรได้ดี ?



Market segmentation (การแบ่งส่วนตลาด)

Organizing computer clusters

(การจัดระเบียบ computer clusters)



Social network analysis

(การวิเคราะห์เครือข่ายทางสังคม)

Astronomical data analysis

(การวิเคราะห์ข้อมูลทางดาราศาสตร์)

Figure source: Andrew Ng's Machine Learning course

Question

ข้อใดต่อไปนี้เป็นจริง? ตอบทุกข้อที่จริง

(i) ใน unsupervised learning ชุดข้อมูล training set มีรูปแบบเป็น $\{x^{(1)}, x^{(2)}, ..., x^{(m)}\}$ โดยไม่มี labels $y^{(i)}$

(ii) Clustering เป็นตัวอย่างหนึ่งของ unsupervised learning

(iii) ใน unsupervised learning เราใช้ชุดข้อมูลที่ไม่มี label (abeled dataset) และใช้มันหา ‘structure’ (โครงสร้าง) ในข้อมูล

(iv) Clustering เป็น unsupervised learning algorithm เพียง algorithm เดียว

Question

ข้อใดต่อไปนี้เป็นจริง? ตอบทุกข้อที่จริง

(i) ใน unsupervised learning ชุดข้อมูล training set มีรูปแบบเป็น $\{x^{(1)}, x^{(2)}, ..., x^{(m)}\}$ โดยไม่มี labels $y^{(i)}$

(ii) Clustering เป็นตัวอย่างหนึ่งของ unsupervised learning

(iii) ใน unsupervised learning เราใช้ชุดข้อมูลที่ไม่มี label (abeled dataset) และใช้มันหา 'structure' (โครงสร้าง) ในข้อมูล

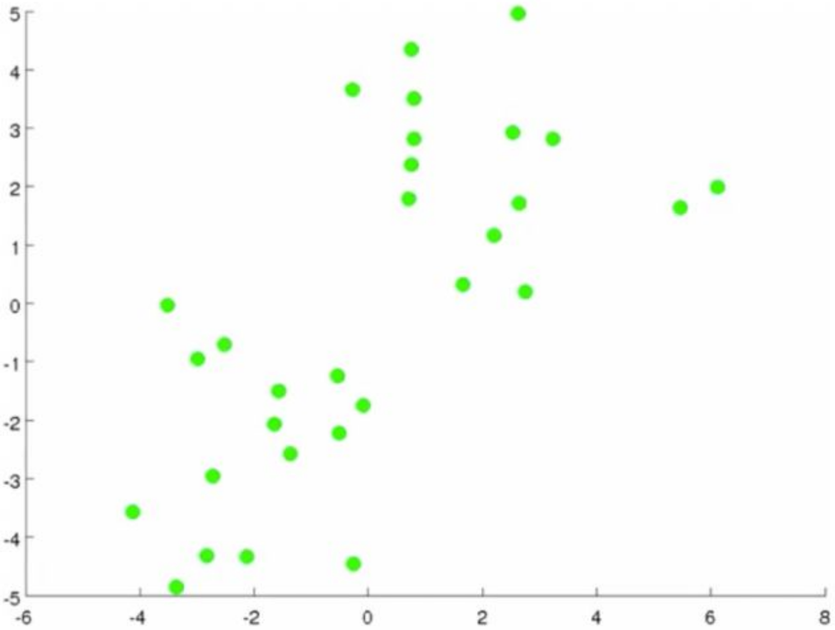
(iv) Clustering เป็น unsupervised learning algorithm เพียง algorithm เดียว

K-means Algorithm

Krittameth Teachasrisaksakul

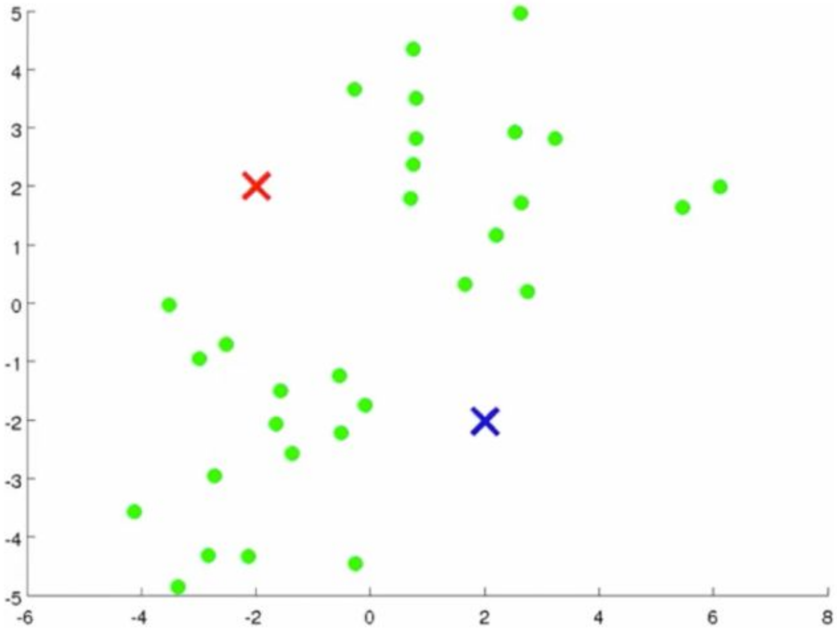
K-means : ความเข้าใจพื้นฐาน

สมมติ เราอยากจัดกลุ่มข้อมูลเป็น สอง cluster (กลุ่ม)



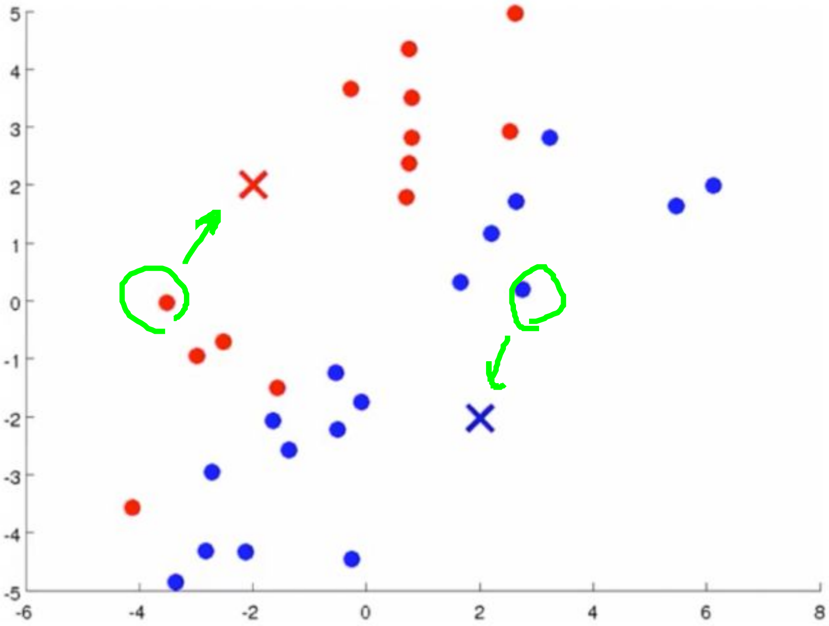
K-means : ความเข้าใจพื้นฐาน

ขั้นที่ 1 : ตั้งค่าเริ่มต้น (initialize) จุด 2 จุด เรียกว่า ‘cluster centroids’



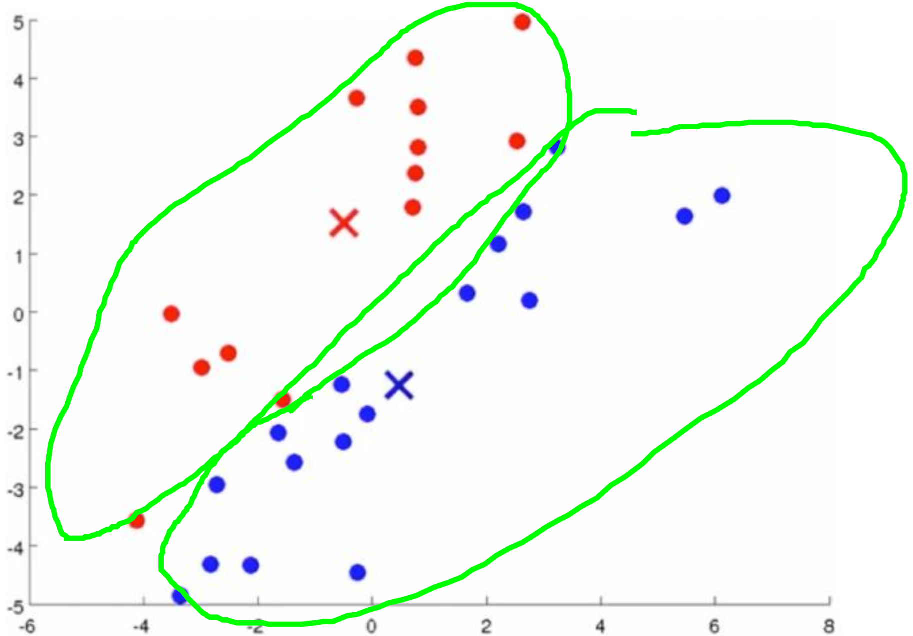
K-means : ความเข้าใจพื้นฐาน

ขั้นที่ 2 : ให้ example แต่ละอัน เป็นสีใดสีหนึ่ง โดยเป็นสีเดียวกับ cluster centroid ที่อยู่ใกล้ที่สุด



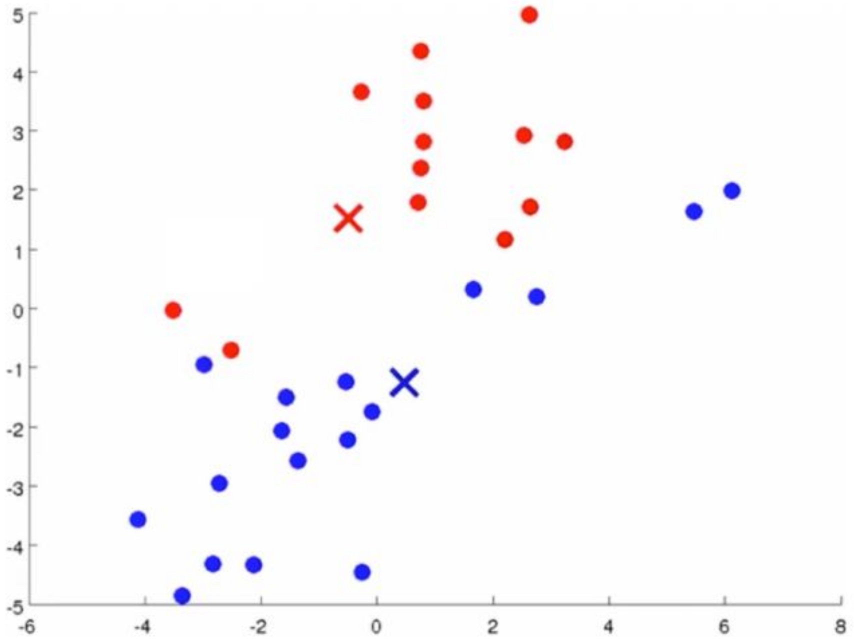
K-means : ความเข้าใจพื้นฐาน

ขั้นที่ 3 : ย้าย cluster centroid ไปอยู่ที่ค่าเฉลี่ยของตัวอย่าง (example) ที่เป็นสีเดียวกัน



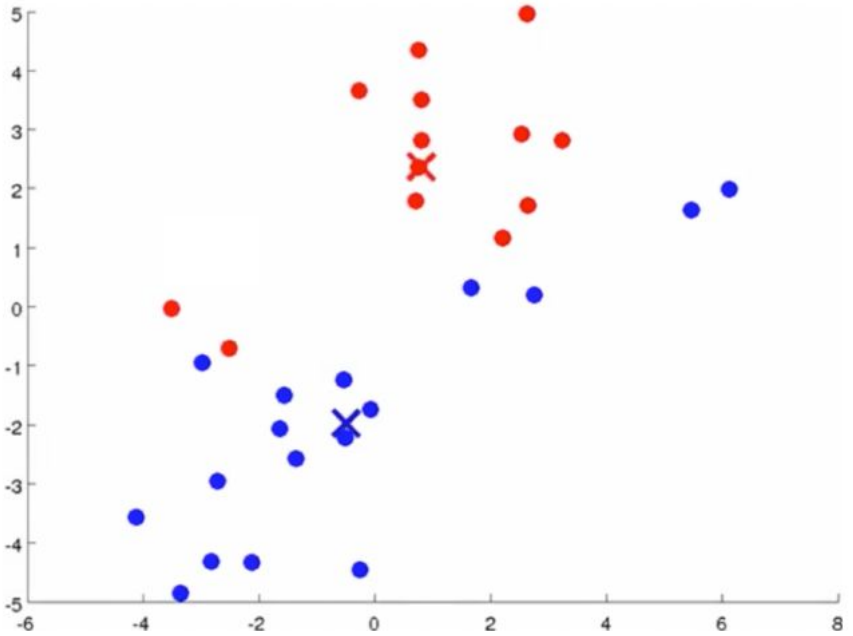
K-means : ความเข้าใจพื้นฐาน

ต่อไป ทำขั้นตอนเดิมซ้ำ (iterative steps) และ จะได้สีของ example ดังนี้



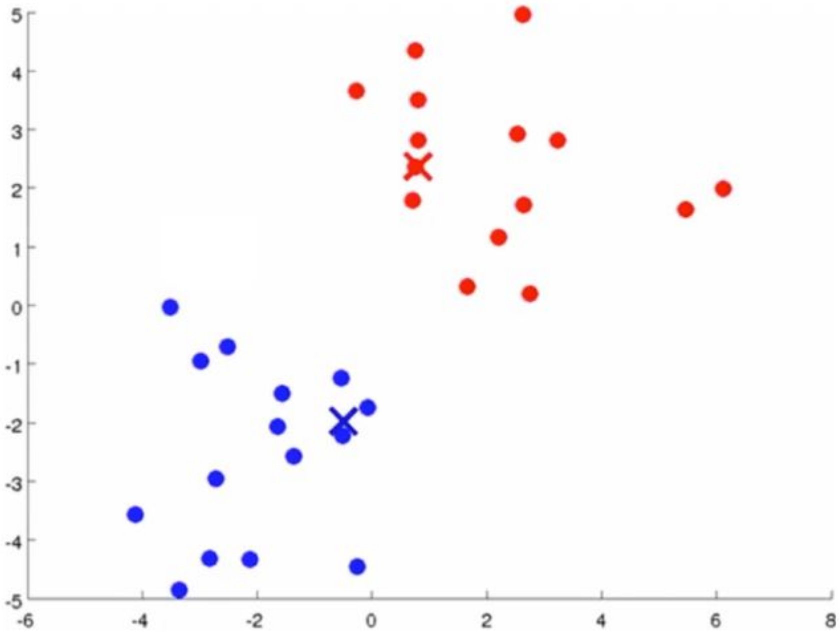
K-means : ความเข้าใจพื้นฐาน

ต่อไป ทำขั้นตอนเดิมซ้ำ (iterative steps) และ จะได้สีของ example ดังนี้



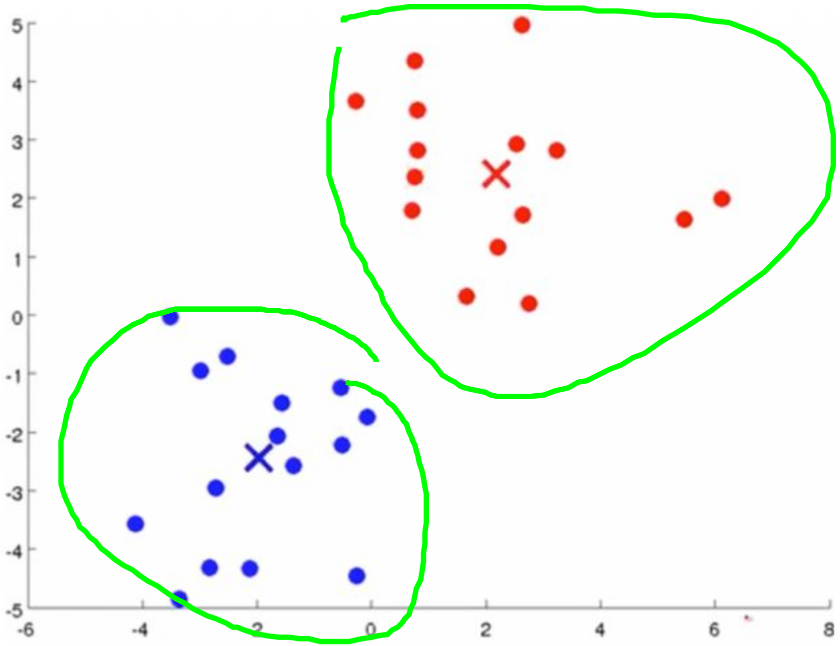
K-means : ความเข้าใจพื้นฐาน

ต่อไป ทำขั้นตอนเดิมซ้ำ (iterative steps) และ จะได้สีของ example ดังนี้



K-means : ความเข้าใจพื้นฐาน

ตอนนี้ K-means ทำงานเสร็จสิ้นแล้ว run อีกหนึ่ง iteration : centroid ก็จะไม่เปลี่ยนแปลง



K-means Algorithm

Input:
- K (จำนวน clusters)
- ชุดข้อมูล training set $\{x^{(1)}, x^{(2)}, ..., x^{(m)}\}$, $x^{(i)} \in \mathbb{R}^n$ (เลิกใช้สัญลักษณ์ $x_0 = 1$)

Algorithm:

ตั้งค่าเริ่มต้น (โดยสุ่ม) → K cluster centroids $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^n$

ทำซ้ำ {

for $i = 1$ to m

$c^{(i)}$:= index (จาก 1 ถึง K) ของ cluster centroid ที่ใกล้ $x^{(i)}$ ที่สุด

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

for $k = 1$ to K

μ_k := ค่าเฉลี่ย (mean) ของจุดที่ถูกจัดให้อยู่ใน cluster k

}

ถ้าไม่มีจุดไหนถูกจัดให้อยู่ใน cluster k ?

(index = เลขดัชนี)

(cluster centroid = จุดกึ่งกลางของ cluster)

Question

สมมติเรา run k-means และหลัง algorithm converge เราได้ว่า: $c^{(1)} = 3,\ c^{(2)} = 3,\ c^{(3)} = 5,\ \ldots$

ข้อใดต่อไปนี้เป็นจริง? วงทุกข้อที่ถูกต้อง

(i) example ที่ 3 $x^{(3)}$ ถูกจัดให้อยู่ใน cluster 5 ✓

(ii) training examples ที่ 1 และ 2 : $x^{(1)}$, $x^{(2)}$ ถูกจัดให้อยู่ใน cluster เดียวกัน ✓

(iii) training examples ที่ 2 และ 3 ถูกจัดให้อยู่ใน cluster เดียวกัน

(iv) จากค่าที่เป็นไปได้ทั้งหมดของ $k \in \{1, 2, \ldots, K\} \rightarrow$ ค่า $k = 3$ ทำให้ $||x^{(2)} - \mu_k||^2$ น้อยที่สุด

Question

สมมติเรา run k-means และหลัง algorithm converge เราได้ว่า: $c^{(1)} = 3, c^{(2)} = 3, c^{(3)} = 5, \ldots$

ข้อใดต่อไปนี้เป็นจริง? วงทุกข้อที่ถูกต้อง

(i) example ที่ 3 $x^{(3)}$ ถูกจัดให้อยู่ใน cluster 5

(ii) training examples ที่ 1 และ 2 : $x^{(1)}$, $x^{(2)}$ ถูกจัดให้อยู่ใน cluster เดียวกัน

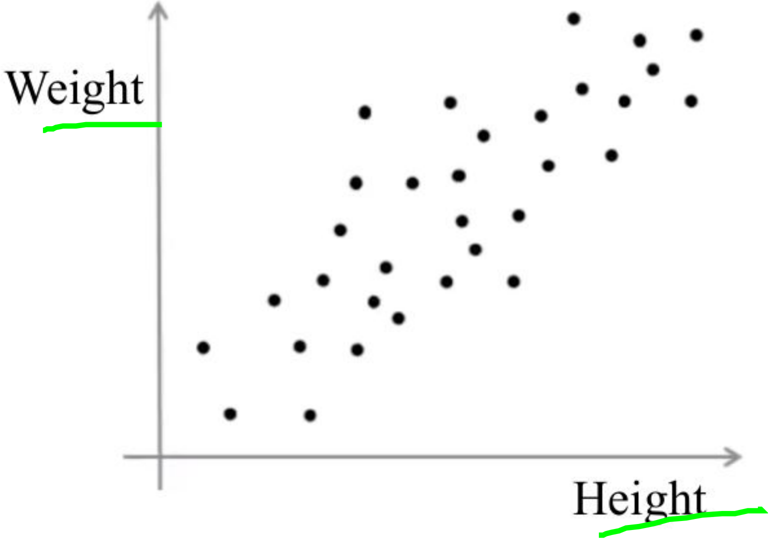
(iii) training examples ที่ 2 และ 3 ถูกจัดให้อยู่ใน cluster เดียวกัน

(iv) จากค่าที่เป็นไปได้ทั้งหมดของ $k \in \{1, 2, \ldots, K\} \rightarrow$ ค่า $k = 3$ ทำให้ $||x^{(2)} - \mu_k||^2$ น้อยที่สุด

k-means สำหรับ Non-Separable Clusters (Cluster ที่แยกไม่ได้อย่างชัดเจน)

Separable Clusters

Non-separable Clusters



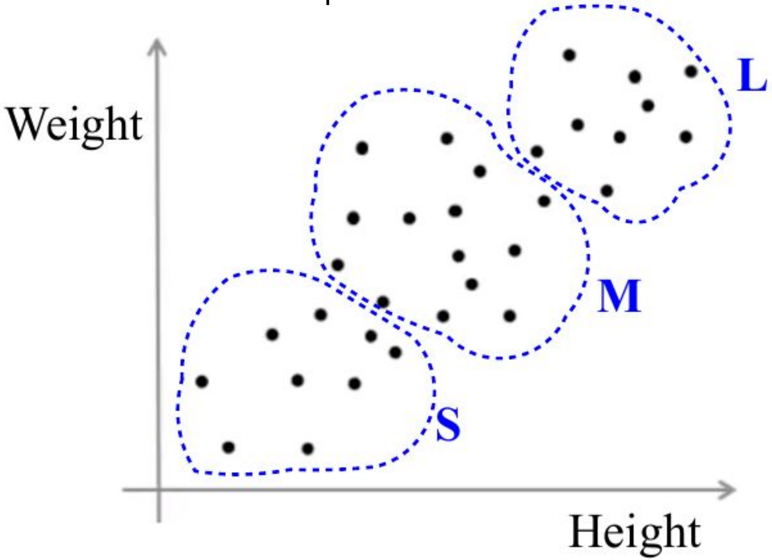
T-shirt Sizing

การหา/แบ่งขนาด T-shirt

k-means สำหรับ Non-Separable Clusters (Cluster ที่แยกไม่ได้อย่างชัดเจน)

Separable Clusters

Non-separable Clusters



T-shirt Sizing

การหา/แบ่งขนาด T-shirt

Optimization Objective

Krittameth Teachasrisaksakul

K-means Optimization Objective

สัญลักษณ์อย่างเป็นทางการ (Formal notation)

- $c^{(i)}$:= index ของ clusters $(1, 2, ..., K)$ ที่ example $x^{(i)}$ ถูกจัดให้อยู่ ตอนนี้
- μ_k := cluster centroid k $(\mu_k \in \mathbb{R}^n)$
- $\mu_{c^{(i)}}$:= cluster centroid ของ cluster ที่ example $x^{(i)}$ ถูกจัดให้อยู่

Cost Function (บางครั้งเรียกว่า 'distortion')

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

Objective Function

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

K-means Algorithm

(index = เลขดัชนี)

(cluster centroid = จุดกึ่งกลางของ cluster)

Input:
- K (จำนวน clusters)
- ชุดข้อมูล training set $\{x^{(1)}, x^{(2)}, ..., x^{(m)}\}$, $x^{(i)} \in \mathbb{R}^n$ (เลิกใช้สัญลักษณ์ $x_0 = 1$)

Algorithm:

ตั้งค่าเริ่มต้น (โดยสุ่ม) K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$

ทำซ้ำ {

(A) for $i = 1$ to m
$c^{(i)} :=$ index

(A) 'Cluster assignment step' (ขั้นตอนจัด example ให้อยู่ใน cluster) เราสามารถแสดงว่ามันทำ:

(จาก 1 ถึง K) ของ k cluster centroid ที่ใกล้ $x^{(i)}$ ที่สุด

(ตัวแปรอื่นๆ มีค่าคงเดิม)

$$\min_{c^{(1)},\ldots,c^{(m)}} J(c^{(1)}, \ldots, c^{(m)})$$

(B) for $k = 1$ to K
$\mu_k :=$ ค่าเฉลี่ย

(B) 'Move centroid step' (ขั้นตอนย้าย centroid)
เราสามารถแสดงว่ามันทำ:

(mean) ของจุดที่ถูกจัดให้อยู่ใน cluster k

$$\min_{\mu_1,\ldots,\mu_K} J(\mu_1, \ldots, \mu_K)$$

(ตัวแปรอื่นๆ มีค่าคงเดิม)

}

Question

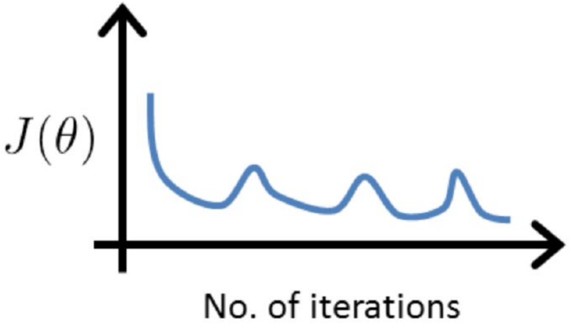
สมมติ เรา implement K-means และเพื่อตรวจสอบว่ามันทำงานอย่างถูกต้อง เรา plot cost function $J(c^{(1)}, ..., c^{(m)}, \mu_1, ..., \mu_K)$ เป็น function ของจำนวน iterations แล้ว plot เป็นแบบด้านล่าง มันหมายความว่าอะไร?

(i) learning rate มากเกินไป

(ii) algorithm ทำงานอย่างถูกต้อง

(iii) algorithm กำลังทำงาน แต่ k มากเกินไป

(iv) บางครั้ง มันเป็นไปไม่ได้ที่จะเพิ่มขึ้น ต้องมี bug ใน code



Question

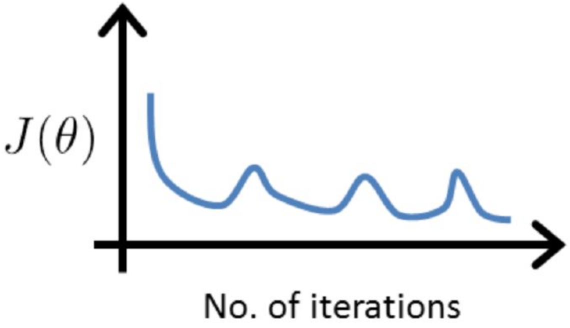
สมมติ เรา implement K-means และเพื่อตรวจสอบว่ามันทำงานอย่างถูกต้อง เรา plot cost function $J(c^{(1)}, ..., c^{(m)}, \mu_1, ..., \mu_K)$ เป็น function ของจำนวน iterations แล้ว plot เป็นแบบด้านล่าง มันหมายความว่าอะไร?

(i) learning rate มากเกินไป

(ii) algorithm ทำงานอย่างถูกต้อง

(iii) algorithm กำลังทำงาน แต่ k มากเกินไป

(iv) บางครั้ง มันเป็นไปไม่ได้ที่จะเพิ่มขึ้น ต้องมี bug ใน code



K-Means

Random Initialization
การตั้งค่าเริ่มต้น โดยสุ่ม

Krittameth Teachasrisaksakul

K-means Algorithm

Input: - K (จำนวน clusters)

- ชุดข้อมูล training set $\{x^{(1)}, x^{(2)}, ..., x^{(m)}\}$, $x^{(i)} \in \mathbb{R}^n$ (เลิกใช้สัญลักษณ์ $x_0 = 1$)

Algorithm:

ตั้งค่าเริ่มต้น (โดยสุ่ม) K cluster centroids $\mu_1, \mu_2, ..., \mu_K \in \mathbb{R}^n$

ทำซ้ำ {

for $i = 1$ to m

$c^{(i)} :=$ index (จาก 1 ถึง K) ของ cluster centroid ที่ใกล้ $x^{(i)}$ ที่สุด

for $k = 1$ to K

$\mu_k :=$ ค่าเฉลี่ย (mean) ของจุดที่ถูกจัดให้อยู่ใน cluster k

}

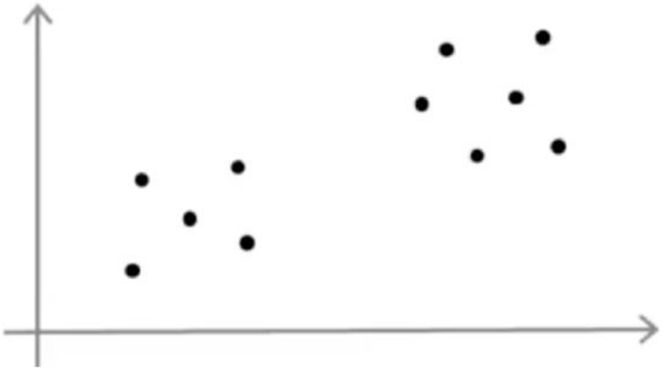
(index = เลขดัชนี)

(cluster centroid = จุดกึ่งกลางของ cluster)

Random Initialization: การตั้งค่าเริ่มต้น โดยสุ่ม

แนวคิด:

- ควรมี $K < m$
- เลือก training example K ตัว แบบสุ่ม
- ตั้งค่า $\mu_1, \ldots, \mu_K$ เป็น example K ตัวนี้

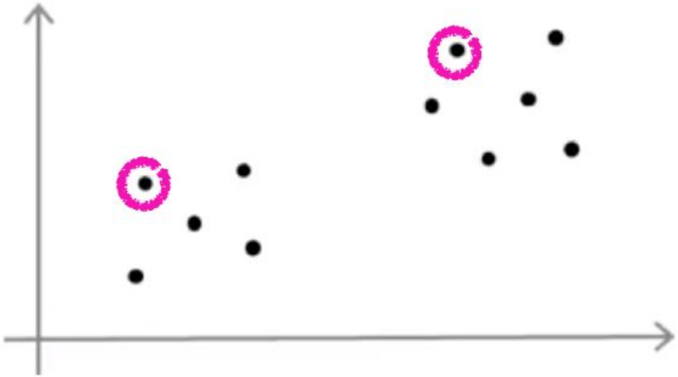


Random Initialization: การตั้งค่าเริ่มต้น โดยสุ่ม

แนวคิด:

- ควรมี $K < m$
- เลือก training example K ตัว แบบสุ่ม
- ตั้งค่า $\mu_1, \ldots, \mu_K$ เป็น example K ตัวนี้

ตัวอย่าง 1: สมมติ $K = 2$

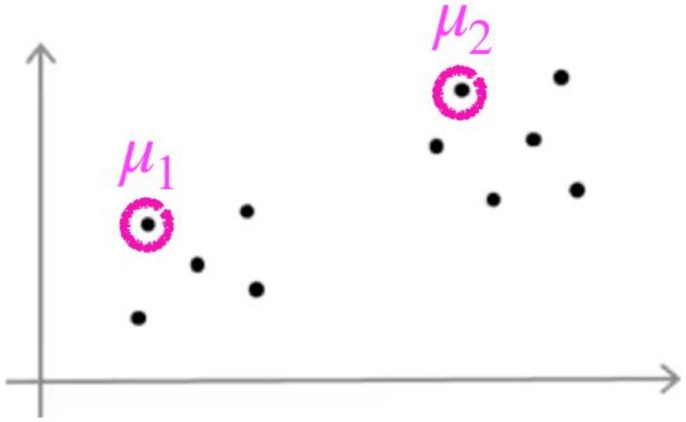


Random Initialization: การตั้งค่าเริ่มต้น โดยสุ่ม

แนวคิด:

- ควรมี $K < m$
- เลือก training example K ตัว แบบสุ่ม
- ตั้งค่า $\mu_1, \ldots, \mu_K$ เป็น example K ตัวนี้

ตัวอย่าง 1: สมมติ $K = 2$

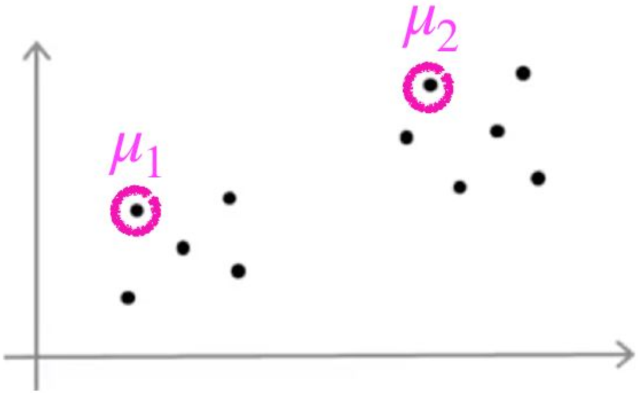


นี่อาจเป็นกรณีที่โชคดี !

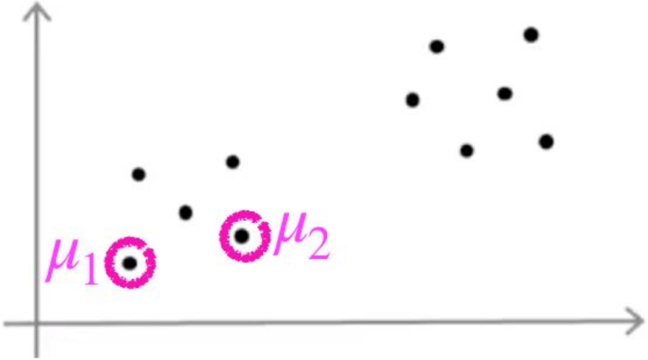
Random Initialization: การตั้งค่าเริ่มต้น โดยสุ่ม

แนวคิด:

- ควรมี $K < m$
- เลือก training example K ตัว แบบสุ่ม
- ตั้งค่า $\mu_1, \ldots, \mu_K$ เป็น example K ตัวนี้



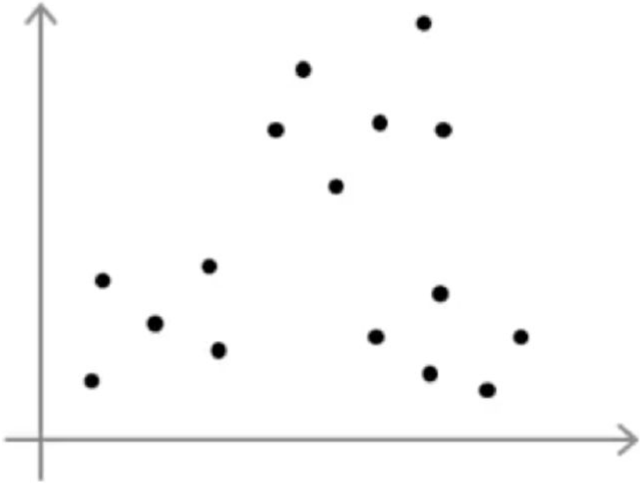
ตัวอย่าง 1: สมมติ $K = 2$



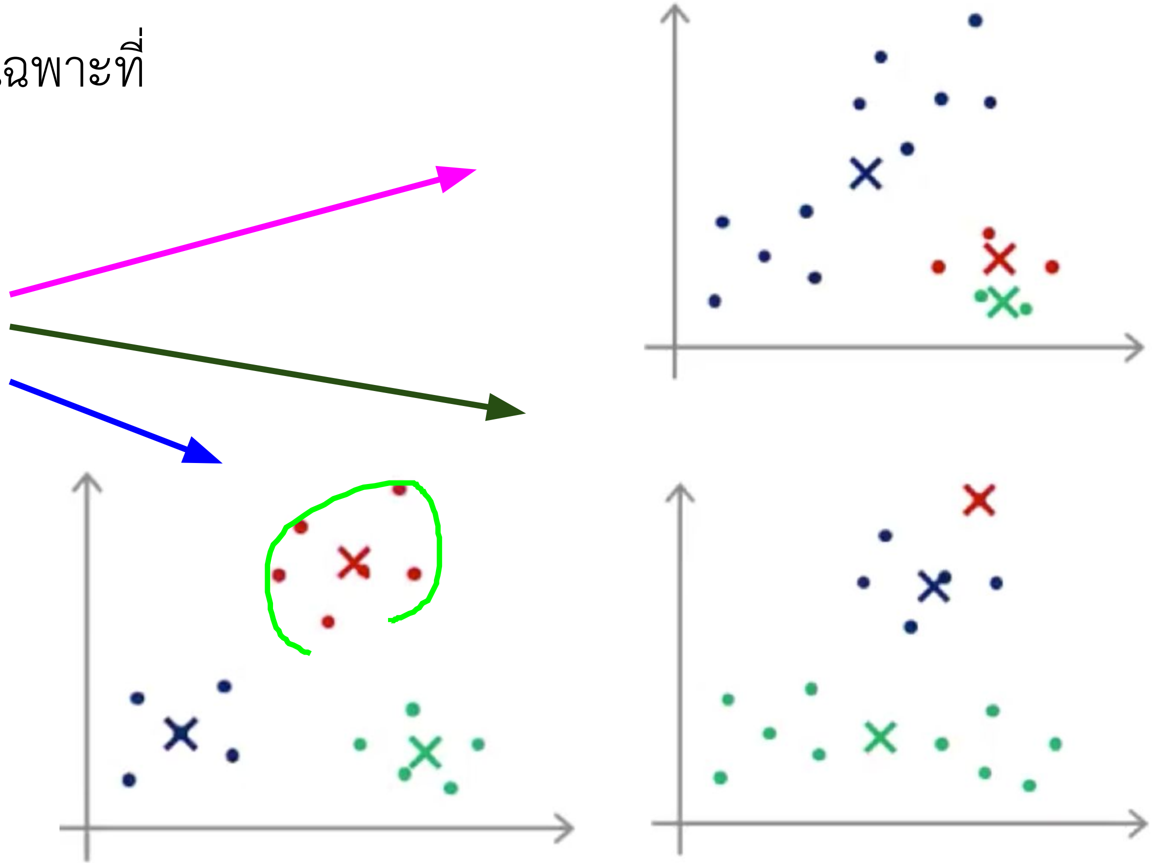
ตัวอย่าง 2: สมมติ $K = 2$

คำเตือน ! *K*-means อาจ converge แล้วได้คำตอบ (solution) ที่ต่างไป ขึ้นอยู่กับว่า $\mu_1, \ldots, \mu_K$ ถูกตั้งค่าเริ่มต้น (initialize) อย่างไร !

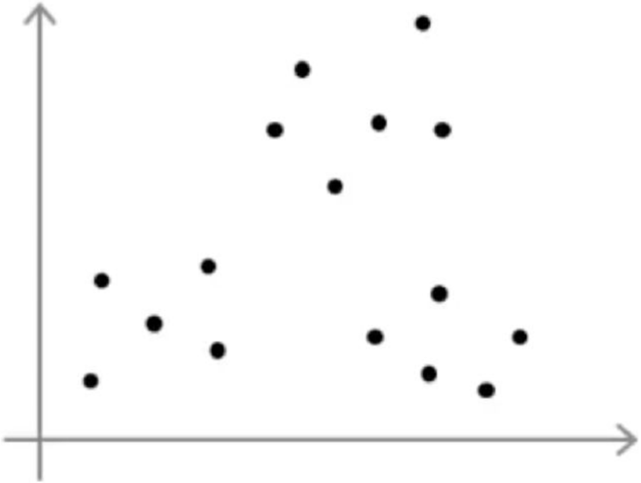
Local Optima : ค่าที่เหมาะสมเฉพาะที่

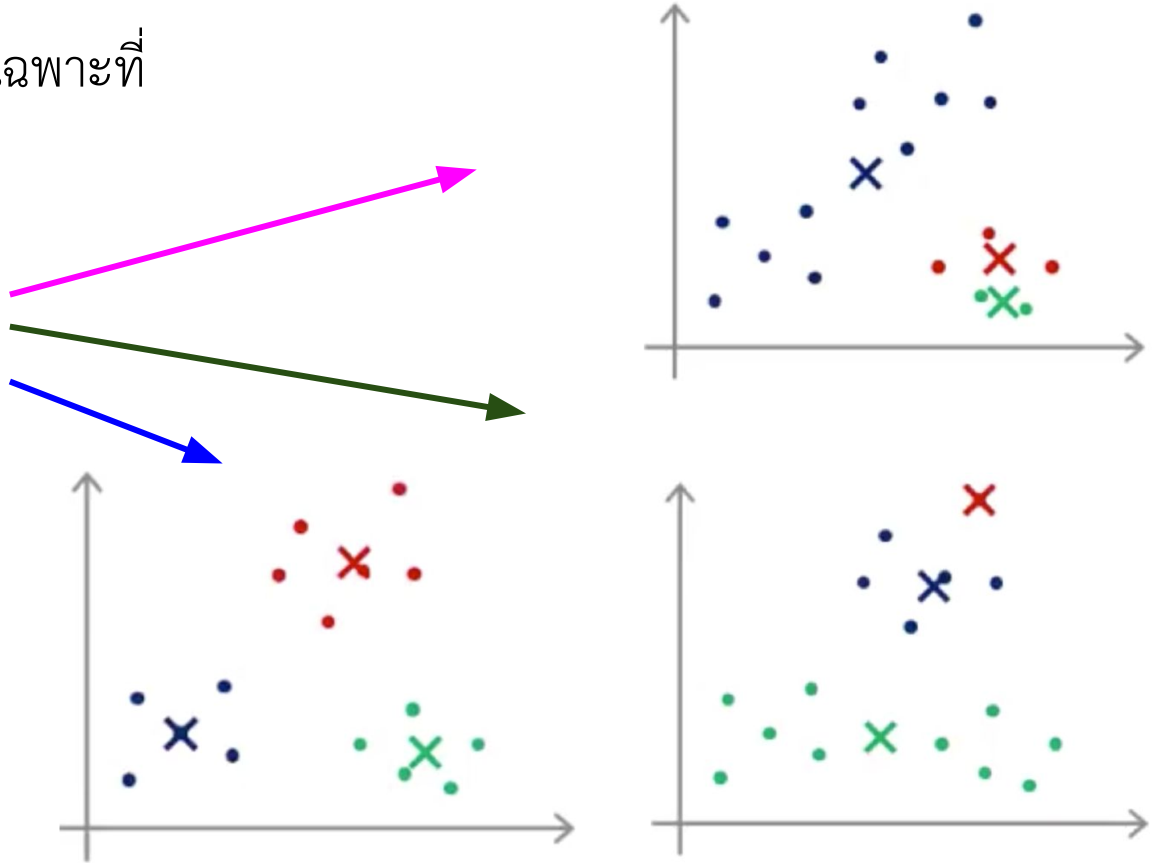


ค่า Local Optima ที่ไม่ดี



Local Optima : ค่าที่เหมาะสมเฉพาะที่





หลีกเลี่ยง การติดอยู่ที่ค่า local optima อย่างไร? (ลอง random initialization หลายแบบ และ run *k*-means หลายครั้ง)

Random Initialization: การตั้งค่าเริ่มต้น แบบสุ่ม

ค่านี้ สามารถมากกว่านี้ เป็น 10 – 1,000 เท่า

for i = 1 to 100 {

ตั้งค่าเริ่มต้น K-means แบบสุ่ม

Run K-means ⟶ ได้ค่า $c^{(1)}, ..., c^{(m)}, \mu_1, ..., \mu_K$

คำนวณ cost function (หรือ distortion)

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

}

เลือก clustering (การแบ่งกลุ่ม) ที่ cost ต่ำที่สุด $J(c^{(1)}, ..., c^{(m)}, \mu_1, ..., \mu_K)$

Random Initialization: การตั้งค่าเริ่มต้น แบบสุ่ม

ค่านี้ สามารถมากกว่านี้ เป็น 10 – 1,000 เท่า

for i = 1 to 100 {

ตั้งค่าเริ่มต้น K-means แบบสุ่ม

Run K-means ⟶ ได้ค่า $c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K$

คำนวณ cost function (หรือ distortion)

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

}

เลือก clustering (การแบ่งกลุ่ม) ที่ cost ต่ำที่สุด $J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$

ถ้าเรา run k-means ด้วยจำนวน cluster ที่ค่อนข้างน้อย (เช่น $K = 2 - 10$) การทำ random initialization หลายแบบ สามารถทำให้ได้ค่า local optima ที่ดีกว่า บางครั้ง ไม่อย่างนั้น การทำ random initialization หลายแบบ อาจทำให้ได้ค่า local optima ที่ดีกว่า แต่ไม่มาก

Question

ข้อใดต่อไปนี้ เป็นวิธีที่แนะนำให้ตั้งค่าเริ่มต้น (initialize) k-means

(i) เลือกจำนวนเต็มโดยสุ่มจาก $\{1, ..., K\}$
ตั้งค่า $\mu_1 = \mu_2 = ... = \mu_K = x^{(i)}$

(ii) เลือกจำนวนเต็ม k ตัวที่ไม่ซ้ำกัน $i_1, ..., i_k$ โดยสุ่มจาก $\{1, ..., K\}$
ตั้งค่า $$\mu_1 = x^{(i_1)}, \mu_2 = x^{(i_2)}, \ldots, \mu_K = x^{(i_k)}$$

(iii) เลือกจำนวนเต็ม k ตัวที่ไม่ซ้ำกัน $i_1, ..., i_k$ โดยสุ่มจาก $\{1, ..., m\}$
ตั้งค่า $$\mu_1 = x^{(i_1)}, \mu_2 = x^{(i_2)}, \ldots, \mu_K = x^{(i_k)}$$

(iv) ตั้งค่าสมาชิกทุกตัวของ $\mu_i \in \mathbb{R}^n$ เป็นค่าที่สุ่ม จากช่วงระหว่าง $-\varepsilon$ และ ε โดย ε เป็นค่าที่น้อย

Question

ข้อใดต่อไปนี้ เป็นวิธีที่แนะนำให้ตั้งค่าเริ่มต้น (initialize) k-means

(i) เลือกจำนวนเต็มโดยสุ่มจาก $\{1, ..., K\}$
ตั้งค่า $\mu_1 = \mu_2 = ... = \mu_K = x^{(i)}$

(ii) เลือกจำนวนเต็ม k ตัวที่ไม่ซ้ำกัน $i_1, ..., i_k$ โดยสุ่มจาก $\{1, ..., K\}$
ตั้งค่า $$\mu_1 = x^{(i_1)}, \mu_2 = x^{(i_2)}, \ldots, \mu_K = x^{(i_k)}$$

(iii) เลือกจำนวนเต็ม k ตัวที่ไม่ซ้ำกัน $i_1, ..., i_k$ โดยสุ่มจาก $\{1, ..., m\}$
ตั้งค่า $$\mu_1 = x^{(i_1)}, \mu_2 = x^{(i_2)}, \ldots, \mu_K = x^{(i_k)}$$

k m

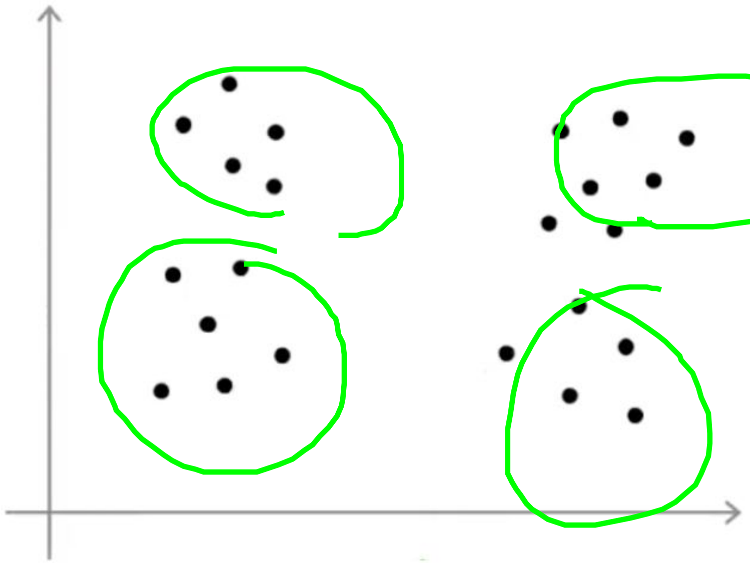
(iv) ตั้งค่าสมาชิกทุกตัวของ $\mu_i \in \mathbb{R}^n$ เป็นค่าที่สุ่ม จากช่วงระหว่าง $-\varepsilon$ และ ε โดย ε เป็นค่าที่น้อย

K-Means

การเลือกจำนวน Clusters

Krittameth Teachasrisaksakul

ค่า K ที่เหมาะสม เป็นเท่าไร?

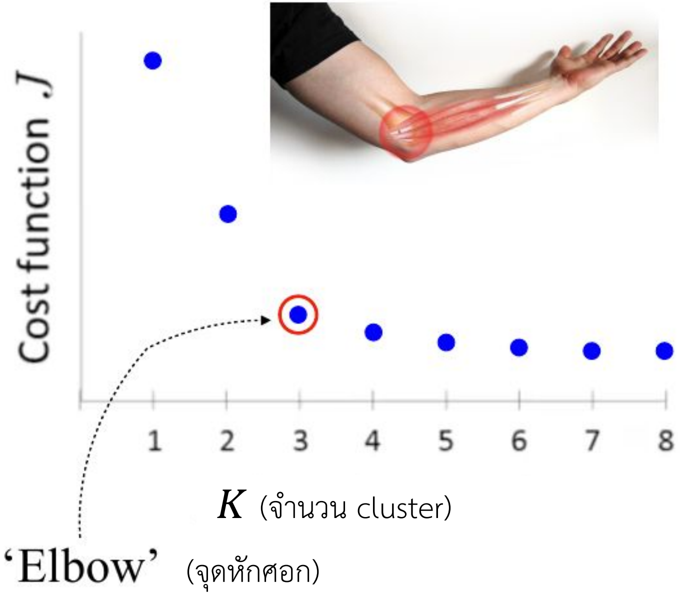


คำถามนี้ค่อนข้าง subjective to คำตอบ
ดูเหมือนว่า ไม่มีค่าใดที่เหมาะสม

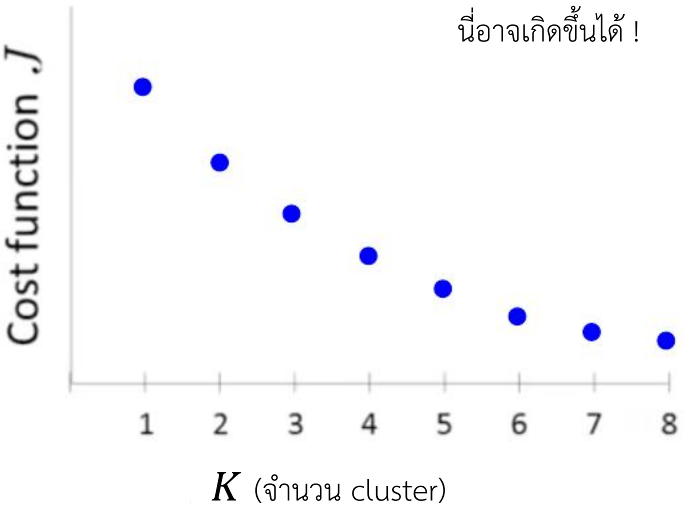
2 or 4 ?

การเลือกค่า K

Elbow method: เปลี่ยนค่า K จาก $1, 2, \ldots$ แล้วเลือกค่า K ที่อยู่ที่จุดหักศอก (elbow)



นี่อาจเกิดขึ้นได้ !



Question

สมมติเรา run K-means ด้วยค่า $K = 3$ และ $K = 5$
แล้วพบว่า cost function J เมื่อ $K = 5$ มีค่าสูงกว่า เมื่อ $K = 3$ มากๆ
เราสามารถสรุปได้ว่าอะไร?

(i) เป็นไปไม่ได้ในทางคณิตศาสตร์ ต้องมี bug ใน code

(ii) จำนวน cluster ที่ถูกต้อง คือ $K = 3$

(iii) เมื่อ run ด้วย $K = 5$: K-means ติดอยู่ที่ค่า local minima ที่ไม่ดี
เราควรลอง run K-means ด้วย random initialization หลายๆแบบ

(iv) เมื่อ run ด้วย $K = 3$: K-means ฟลุค เราควรลอง run K-means อีกครั้ง ด้วย $K = 3$ และ random initialization แบบที่ต่างไป จนกระทั่ง มันทำงานได้แย่กว่าเมื่อ $K = 5$

Question

สมมติเรา run K-means ด้วยค่า $K = 3$ และ $K = 5$
แล้วพบว่า cost function J เมื่อ $K = 5$ มีค่าสูงกว่า เมื่อ $K = 3$ มากๆ
เราสามารถสรุปได้ว่าอะไร?

(i) เป็นไปไม่ได้ในทางคณิตศาสตร์ ต้องมี bug ใน code

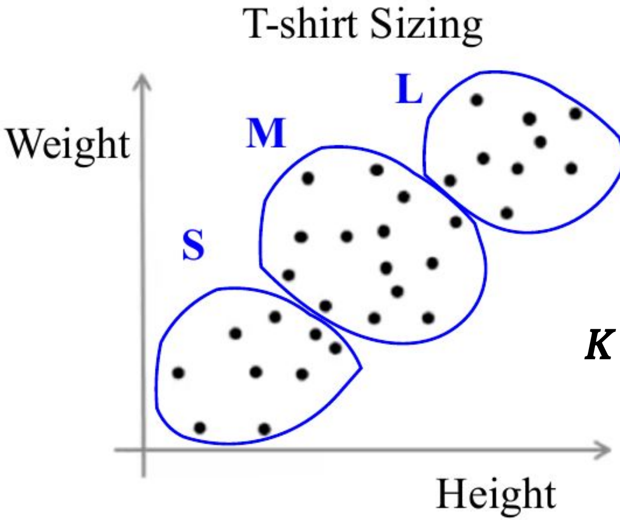
(ii) จำนวน cluster ที่ถูกต้อง คือ $K = 3$

(iii) เมื่อ run ด้วย $K = 5$: K-means ติดอยู่ที่ค่า local minima ที่ไม่ดี
เราควรลอง run K-means ด้วย random initialization หลายๆแบบ

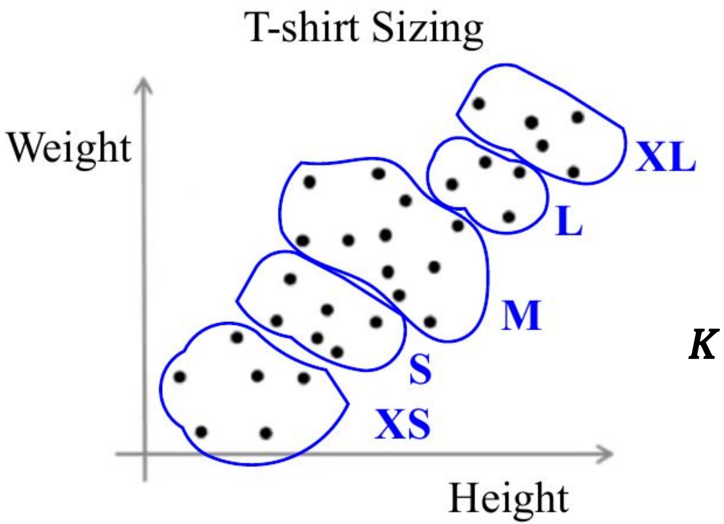
(iv) เมื่อ run ด้วย $K = 3$: K-means ฟลุค เราควรลอง run K-means อีกครั้ง ด้วย $K = 3$ และ random initialization แบบที่ต่างไป จนกระทั่ง มันทำงานได้แย่กว่าเมื่อ $K = 5$

การเลือกค่า K

บางครั้ง เรา run *k*-means เพื่อหา cluster เพื่อจะใช้ทำงานบางอย่างที่ปลายทาง (downstream purpose) ภายหลัง เราจึงต้องประเมิน *K*-means โดยใช้ metric (ตัววัด) ว่ามันทำงานได้ดีแค่ไหนเมื่อใช้มันทำงานนั้นๆ เช่น T-shirt sizing (การหาขนาดที่เหมาะสมของ T-shirt แต่ละ size) เช่น



$K = 3$



$K = 5$

สรุป

- บ่อยครั้ง ค่า K จะถูกเลือกด้วยมือ
- วิธีหนึ่งเพื่อเลือกค่า K คือ Elbow method
  แต่มันอาจไม่ทำงานได้ดีเสมอไป
- วิธีที่เหมาะสมมากกว่า คือ ถามว่า:
  "เราจะ run *k*-means ไปเพื่อทำงานอะไร ?"
  แล้วเลือกค่า K ที่ทำให้ทำงานนั้นได้ดี

T

C

T

C

T

C

C

References

1. Andrew Ng, Machine Learning, Coursera.
2. Teeradaj Racharak, AI Practical Development Bootcamp.
3. What is Machine Learning?, https://www.digitalskill.org/contents/5

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 1.2]
Figure source: Andrew Ng's Machine Learning course