การบ้านการเขียนโปรแกรม 7:
K-means Clustering

Thanks Andrew Ng for this beautiful programming exercise

ในการบ้านนี้เราจะทดลองเขียนโปรแกรมเพื่อบีบอัดภาพโดยใช้ K-means Clustering ด้วย Octave/Matlab

ในpackage ประกอบด้วยไฟล์
ส่วนแรก

- ex7.m - สคริปต์เพื่อรันโปรแกรม
- ex7data2.mat - ชุดข้อมูล 1
- bird_small.png - ตัวอย่างภาพนก
- displayData.m - แสดงข้อมูลที่เก็บอยู่ในเมตริกซ์
- plotDataPoints.m - เริ่มต้นค่าจุดศูนย์กลางให้กับ K-means
- plotProgresskMeans.m - พล็อตกราฟแต่ละขั้นตอนของ K-means
- runkMeans.m - รันอัลกอริทึม K-means
- findClosestCentroids.m* - หาจุดศูนย์กลางที่ใกล้ที่สุด
- computeCentroids.m* - คำนวณค่าเฉลี่ยของศูนย์กลาง
- kMeansInitCentroids.m* - กำหนดค่าเริ่มต้นให้กับจุดศูนย์กลาง

* คือ ไฟล์ที่ต้องแก้ไขและส่ง

ตลอดการทดสอบโค้ดของการบ้านครั้งนี้ ส่วนแรกให้สั่งรัน ex7.m

1. K-means Clustering

ในการบ้านนี้คุณต้องเขียนโค้ดเพื่อสร้างอัลกอริทึม K-means และใช้ในการบีบอัดรูปภาพ คุณเริ่มจากข้อมูล 2 มิติในชุดข้อมูลที่ให้มาเพื่อให้คุณเข้าใจหลักการของอัลกอริทึม K-means หลังจากนั้นเราจะใช้ K-means เพื่อบีบอัดภาพด้วยการลดจำนวนของสีในภาพลงเหลือเพียงสีที่พบได้ส่วนมากในภาพนั้น

1.1 Implementing K-means

K-means สามารถจัดข้อมูลที่อยู่คล้ายคลึงกันอยู่ในกลุ่มเดียวกันแบบอัตโนมัติ สมมติว่าเราให้ชุดข้อมูล $\{x^{(1)}, \dots, x^{(m)}\}$ และต้องการรวมข้อมูลเป็น 2-3 กลุ่ม ซึ่ง K-means เป็นอัลกอริทึมที่ทำซ้ำเพื่อเดาจุดศูนย์กลางและจัดข้อมูลเข้าไปในกลุ่มใดกลุ่มหนึ่ง ทำซ้ำด้วยการปรับการเดาไปเรื่อยๆจนกระทั่งได้จุดศูนย์กลางทั้งหมด อัลกอริทึมแบ่งได้เป็น 2 ส่วนหลักๆ ดังนี้

1.1.1 หาจุดศูนย์กลางที่ใกล้ที่สุด

ในส่วนนี้ K-means ทำหน้าที่กำหนดจุดศูนย์กลางที่อยู่ใกล้ชุดข้อมูล $x^{(i)}$ มากที่สุดให้กับ $c^{(i)}$ โดยมีข้อมูลเริ่มต้นคือจุดศูนย์กลางทั้งหมด
สำหรับตัวอย่างข้อมูลทุกชุดเราจะกำหนดให้

$$c^{(i)} = j \text{ ซึ่งทำให้ค่า } ||x^{(i)} - \mu_j||^2 \text{ มีค่าน้อยที่สุด}$$

โดยที่ μ_j คือค่าเฉลี่ยของจุดศูนย์กลางที่ j

คุณต้องแก้ไขโค้ดในไฟล์ findClosetCentroids.m ฟังก์ชันนี้รับอินพุตคือข้อมูลในเมตริกซ์ X และตำแหน่งของจุดศูนย์กลางทั้งหมดอยู่ในตัวแปร centrioids ฟังก์ชันทำหน้าที่คำนวณค่า $c^{(i)}$ ซึ่งคือเมตริกซ์ idx ในโค้ด

คุณสามารถใช้ for loop เพื่อเขียนโค้ดส่วนนี้ได้ เมื่อคุณเขียนโค้ดเสร็จผลลัพธ์ที่ควรได้คือ [1 3 2] ซึ่งคือจุดศูนย์กลางที่กำหนดให้ 3 ตัวอย่างแรก

1.1.2 คำนวณค่าจุดศูนย์กลาง

หลังจากเราได้กำหนดจุดศูนย์กลางให้กับข้อมูลตัวอย่างทุกอันแล้ว ขั้นตอนถัดไปคือการรัน K-means ใหม่ด้วยการหาจุดศูนย์กลางของแต่ละกลุ่มใหม่จากค่าเฉลี่ยของข้อมูลทุกอันที่อยู่ในกลุ่มเดียวกันตามสมการ

$$\mu_k := \frac{1}{|C_k|} \sum_{i \in C_k} x^{(i)}$$

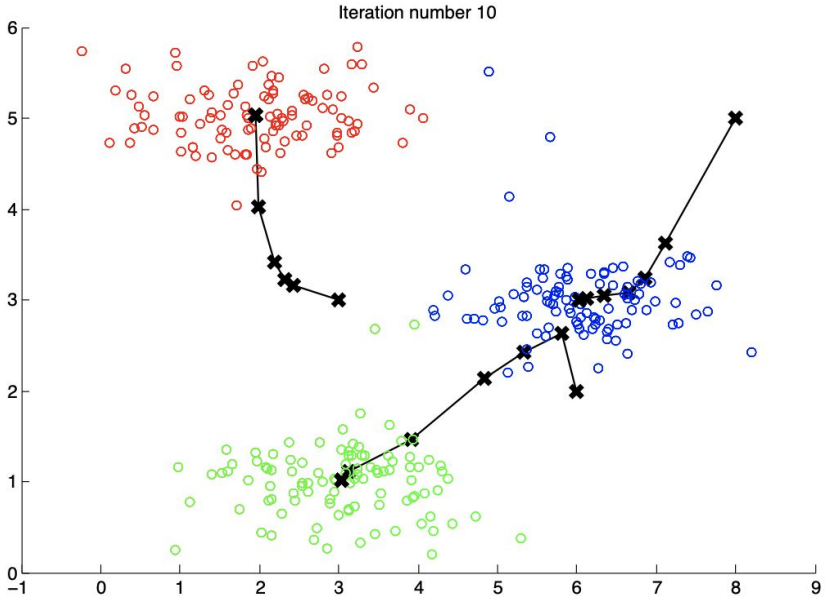
โดยที่ C_k คือกลุ่มของข้อมูลตัวอย่างที่ถูกจัดให้อยู่ในกลุ่ม K ตัวอย่างเช่น ถ้ากลุ่มแรกมี $x^{(1)}$ และ $x^{(2)}$ อยู่ในกลุ่ม $\mu_1 = ½(x^{(1)}+x^{(2)})$

ในส่วนการคำนวณค่าเฉลี่ยให้แก้ไขไฟล์ computeCentroids.m คุณสามารถใช้ for loop ได้

1.2 K-means บนข้อมูลตัวอย่าง

หลังจากคุณแก้ไขโค้ดทั้งสองส่วนเสร็จ สคริปต์จะรันโปรแกรมบนข้อมูล 2D เพื่อง่ายต่อการเข้าใจการทำงานของ K-means กระบวนการทั้งหมดอยู่ในไฟล์ runKmeans.m (นิสิตสามารถศึกษาให้เข้าใจขั้นตอนของอัลกอริทึมได้ที่ไฟล์นี้)

เมื่อโปรแกรมรันผ่านขั้นตอนแรก จะมาถึงขั้นตอนที่สอง ในส่วนนี้โปรแกรมจะแสดงกราฟการเป็นไปของ K-means ในแต่ละรอบ เมื่อรันครบ จะปรากฏภาพดังรูปที่ 1



รูปที่ 1 ผลลัพธ์ของ K-means

เมื่อโปรแกรมรันผ่านขั้นตอนที่สอง จะเข้าสู่การบีบอัดภาพ (ภาพเริ่มต้นสามารถดูได้จากไฟล์ bird_small.png) เมื่อเรารัน K-means จากไฟล์ภาพสี 24-bit ที่แต่ละบิตเก็บ 8-bit ของ RGB จะถูกยุบเหลือแค่ 16 สี โดยที่แต่ละบิตจะถูกแทนที่ด้วย index ของสีที่จะแสดง ณ ตำแหน่งนั้น
16 สีที่ว่านี้จะมาจากการหาตัวแทนของสีด้วยอัลกอริทึม K-means

ตารางคะแนน

ที่	งานที่ต้องทำ	ไฟล์ที่ต้องแก้ไข	คะแนน
1	หาจุดศูนย์กลางที่ใกล้ที่สุด	findClosestCentroids.m	25
2	คำนวณค่าเฉลี่ยของจุดศูนย์กลาง	computeCentroids.m	25
	รวมคะแนน		50