

การบ้านการเขียนโปรแกรม 1:

Linear Regression

Thanks Andrew Ng for this beautiful programming exercise

ในการบ้านนี้เราจะทดลองเขียนโปรแกรมเพื่อทำตามขั้นตอนของ linear regression โดยใช้ matrix operation ด้วย Octave/Matlab

ในpackage ประกอบด้วยไฟล์

- ex1.m - เป็น main script ของการเรียกใช้ฟังก์ชันต่างๆ (ไม่ต้องแก้ไขอะไรในนี้ แต่ให้เรียก ex1 เพื่อรัน)
- ex1data1.txt - เป็นชุดข้อมูลเพื่อใช้สอน และจะเรียกใช้เพื่อสร้างการทำนายด้วย linear regression
- *warmUpExercise.m - ฟังก์ชันเพื่อสร้างความคุ้นเคยกับการใช้งานเมตริกซ์และ Octave/Matlab environment
- plotData.m - ฟังก์ชันเพื่อแสดงกราฟ
- *computeCost.m - ฟังก์ชันเพื่อคำนวณ cost
- *gradientDescent.m - ฟังก์ชันที่ใช้ทำ Parameter learning ด้วยวิธี gradient descent

* คือ ไฟล์ที่ต้องแก้ไขและส่ง

วิธีส่งงาน

1. ให้ลบส่วนที่คุณเขียนเกินและไม่จำเป็นออกทั้งหมดก่อนส่ง และเขียนโค้ดเดิมลงในขอบเขตที่กำหนดให้เท่านั้น ดังภาพ

```
% ===== YOUR CODE HERE =====  
% Instructions: Compute the cost of a particular choice of theta  
% You should set J to the cost.  
%
```

```
% =====
```

*****อาจารย์จะไม่ตรวจบรรทัดอื่นๆนอกขอบเขตนี้ *****

2. อัปโหลดเฉพาะไฟล์ที่กำหนดเท่านั้น

1. การโปรแกรม matrix อย่างง่าย (10 คะแนน)

การบ้านส่วนนี้ให้คุณแก้ไข warmUpExercise.m เพื่อให้โปรแกรมแสดงเอาท์พุต เป็นเมตริกซ์เอกลักษณ์ขนาด 3x3 ซึ่งทำได้โดยโค้ดดังต่อไปนี้

```
A=eye(3);
```

ถ้าต้องการทดสอบให้รันที่ ex1.m หรือพิมพ์ ex1 ที่ Octave console

สิ่งที่โปรแกรมต้องแสดงผลคือ

```
1 0 0
0 1 0
0 0 1
```

2. การโปรแกรม linear regression with one variable

ในส่วนนี้ประกอบด้วย 2 ส่วนย่อยคือ cost function และ gradient descent การที่เราจะสร้างโปรแกรม linear regression ได้ต้องมีส่วนสำคัญอื่นๆ (ที่ไม่ได้ต้องทำเป็นการบ้าน) ซึ่งใน package มีส่วนที่ทำหน้าที่นั้นให้คุณเสร็จแล้ว คุณอาจศึกษาเพิ่มเติมได้ เช่น การโหลดข้อมูล อยู่ใน ex1.m ดังภาพ

```
data = load('ex1data1.txt');           % read comma separated data
X = data(:, 1); y = data(:, 2);
m = length(y);                         % number of training examples
```

การแสดงกราฟในฟังก์ชัน plotData ดังภาพ

```
plot(x, y, 'rx', 'MarkerSize', 10);    % Plot the data
ylabel('Profit in $10,000s');           % Set the y-axis label
xlabel('Population of City in 10,000s'); % Set the x-axis label
```

ส่วนต่างๆที่ว่ามานี้ อาจมีประโยชน์ในการปรับใช้เมื่อคุณสร้างระบบของตนเองในอนาคตได้

2.1 การคำนวณ gradient descent

การบ้านนี้ยังเหลืออีก 2 ส่วนย่อยคือ cost function และ gradient descent ทั้ง 2 อันแยกกันไปคนละไฟล์ สิ่งที่คุณต้องทำคือ

2.1.1 cost function (40 คะแนน)

ชื่อไฟล์ที่ต้องแก้ไขชื่อ computeCost.m เป้าหมายคือเพื่อคำนวณค่า cost ดังสมการ

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

โดยที่ คำนวณ hypothesis function ได้จาก

$$h_{\theta}(x) = \theta^T x = \theta_0 + \theta_1 x_1$$

ให้แก้ไขโปรแกรม โดยให้โปรแกรมสามารถคำนวณค่า J ออกมาให้ถูกต้อง ค่าที่คาดหวังคือ J=32.07 ข้อบังคับ คุณจะต้องเขียนโปรแกรมโดยใช้ matrix operation ไม่ใช่การวนลูป ทดสอบด้วยการสั่งรันที่ ex1.m หรือพิมพ์ ex1 ที่ Octave console

2.1.2 gradient descent (50 คะแนน)

ชื่อไฟล์ที่ต้องเขียนส่วนของ gradient descent คือ gradientDescent.m ฟังก์ชันนี้ต้องการให้มันสามารถคำนวณตามหลักการของ gradient descent ให้ถูกต้อง

ทำซ้ำจนกว่าจะลู่เข้า{

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad (\text{simultaneously update } \theta_j \text{ for all } j).$$

}

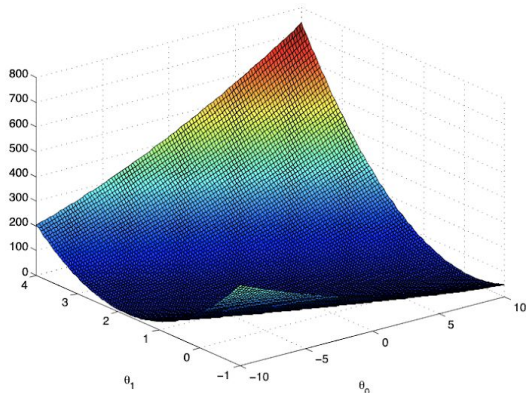
ซึ่งในการโปรแกรมจริง มักจะคำนวณตามจำนวนรอบที่กำหนด ใน ex1.m ได้มีการกำหนดจำนวนรอบไว้เรียบร้อยแล้ว ดังภาพ

```
% Some gradient descent settings
iterations = 1500;
alpha = 0.01;
```

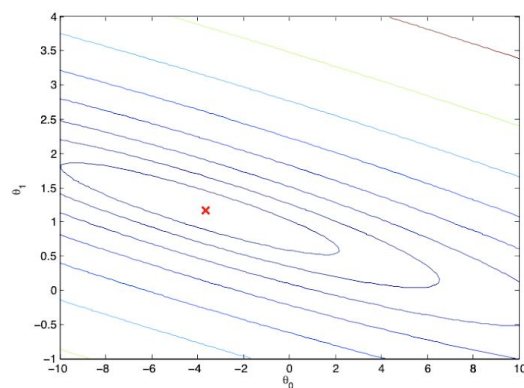
สิ่งที่ต้องคุณต้องทำคือ คำนวณค่า theta ใหม่และปรับค่าในทุกๆรอบ

ข้อบังคับ คุณจะต้องเขียนโปรแกรมโดยใช้ matrix operation ไม่ใช้การวนลูป

คุณสามารถตรวจสอบว่าเขียนโปรแกรมได้ถูกต้อง โดยทดลองแสดงผลลัพธ์ค่า J ซึ่งจะลดลงทุกๆรอบของการรัน นอกจากนี้ เมื่อสั่งรัน ผลที่ได้ควรปรากฏภาพ คล้ายกับด้านล่างนี้



(a) Surface



(b) Contour, showing minimum

3. Linear Regression with Multiple Variables (Optional)

ส่วนนี้เป็นโปรแกรมโดยเพิ่มจำนวน feature ให้มากขึ้น จากส่วนที่ 2 ที่มีเพียง 1 feature เท่านั้น โปรแกรมส่วนนี้ใช้ไฟล์ ex1_multi.m เป็นสคริปต์หลัก ให้ทดสอบและสั่งรันจากไฟล์นี้ และเรียกข้อมูลจาก ex1data2.txt ซึ่งมีข้อมูลเพื่อทำนายราคาบ้านจากพื้นที่ และจำนวนห้องในบ้าน

3.1 feature scaling

เนื่องจาก พื้นที่ และ จำนวนห้อง มีช่วงของข้อมูลที่แตกต่างกันมาก เราจึงต้องทำสเกล ให้ชุดข้อมูลอยู่ในช่วงเดียวกัน

ขั้นตอนนี้ใช้ไฟล์ featureNormalize.m สิ่งที่คุณต้องทำคือ หาค่าเฉลี่ย และส่วนเบี่ยงเบนมาตรฐานแยกแต่ละ feature แล้วใช้ทั้งสองค่านี้ ปรับค่าแต่ละ feature การปรับค่าใช้สูตร

$$x_i = \frac{x_i - \mu}{s_i}$$

แนะนำ std(), mean() สามารถใช้เพื่อหาค่าส่วนเบี่ยงเบนมาตรฐานและ ค่าเฉลี่ยได้
ข้อบังคับ ให้ใช้การคำนวณด้วย matrix ไม่ใช่ loop

3.2 gradient descent

สำหรับการสอนพารามิเตอร์ด้วย gradient descent นั้นสามารถประยุกต์ใช้ gradient descent ในตอน one variable (ข้อที่ 2) ได้ แต่ต้องปรับให้มันไม่ยึดติดกับขนาดใดขนาดหนึ่ง
 ใช้ไฟล์ gradientDescentMulti.m และ computeCostMulti.m
ข้อบังคับ ให้ใช้การคำนวณด้วย matrix ไม่ใช่ loop

3.3 Normal Equation

ใช้สำหรับการทำ gradient descent เช่นกัน แต่เป็นขั้นตอนเดียวแล้วได้คำตอบเลย ไม่ต้องทำซ้ำเพื่อปรับพารามิเตอร์ไปเรื่อยๆ normal equation เป็นดังสมการ

$$\theta = (X^T X)^{-1} X^T y$$

ให้ใช้ไฟล์ normalEqn.m
ข้อบังคับ ให้ใช้การคำนวณด้วย matrix ไม่ใช่ loop

ตารางคะแนน

ที่	งานที่ต้องทำ	ไฟล์ที่แก้ไขและส่ง	คะแนน
1	warm up	warmUpExercise.m	10
2	คำนวณ cost สำหรับ 1 ตัวแปร	computeCost.m	40
3	คำนวณ gradient descent สำหรับ 1 ตัวแปร	gradientDescent.m	50
	รวม		100

คะแนนพิเศษ

ที่	งานที่ต้องทำ	ไฟล์ที่แก้ไขและส่ง	คะแนน
-----	--------------	--------------------	-------

1	feature normalization	featureNormalize.m	5
2	คำนวณ cost สำหรับหลายตัวแปร	computeCostMulti.m	20
3	คำนวณ gradient descent สำหรับหลายตัวแปร	gradientDescentMulti.m	20
4	คำนวณ normal equation	normalEqn.m	5
	รวม		50