ML: HW1

R05922128 林瑋柔

Code: linear regression function by Gradient Descent

```
154 def GD_Regression(D, regRate, GDpara):
155     # model y= sum_item(w[item]x_item)
156     print 'Training...'
157     dimx = len(D[0]['x'])
158     eta, itr, init = GDpara.eta, GDpara.itr, GDpara.init
159 
160     w = np.zeros( dimx, dtype='float' )
161     G = np.zeros( dimx, dtype='float' )
162     X = D['x']
163     y = D['y']
164     w[ itemIdx['PM2.5'] ] = init #initial value
165     if itr>0:
166         for i in range(int(itr)):
167             g = -2* np.dot(( y-np.dot(X,w)), X) + 2*regRate*w
168             G += g*g
169             w = w-eta*g/np.power(G,0.5)
170             print i, meanErr(D,w), reg(w)
171     else:
172         stopRate = -1*itr
173         w_last = np.zeros(dimx, dtype='float')
174         changeRate = stopRate+1
175         while changeRate >= stopRate:
176             g = -2* np.dot(( y-np.dot(X,w)), X) + 2*regRate*w
177             G += g*g
178             change = eta*g/np.power(G,0.5)
179             w = w-change
180             changeRate = np.linalg.norm(change)/np.linalg.norm(w)
181             print changeRate, meanErr(D,w), reg(w)
182     print 'Complete training.^G\n'
183     return w
```

D.dtype=[('x', str(dimx)+'float'),('y', 'float')]

x[0]為常數項（全部都是 1）

itr: Gradient Descent 的 iteration 數

可指定 iteration 或者 w 改變夠小時停下來

g: gradient of the Loss function

G: AdaGrad 的係數

Method

Feature 有一次項、二次項、三次項、exponential、乘積等等，用 4-fold validation 判斷一組 feature 的好壞，一直改變 feature 看哪個最好。最後使用的兩個 Model：

時間	-7	-6	-5	-4	-3	-2	-1
PM2.5	x	x	x	x	x	x^2,x	x^2,x
AMB_TEMP	x	x	x	x	x	x^2,x	x^2,x
NO	x	x	x	x	x	x^2,x	x^2,x
NO2	x	x	x	x	x	x^2,x	x^2,x
NOx	x	x	x	x	x	x^2,x	x^2,x
O3	x	x	x	x	x	x^2,x	x^2,x
PM10	x	x	x	x	x	x^2,x	x^2,x
THC	x	x	x	x	x	x^2,x	x^2,x
Validate Average Error							**4.24**

時間	-9	-8	-7	-6	-5	-4	-3	-2	-1
PM2.5	x,x^2	x,x^2	x,x^2	x,x^2	x,x^2	x,x^2	x,x^2	x,x^2	x,x^2
CH4							x	x	x
CO							x	x	x
NMHC							x	x	x
NO							x	x	x
NO2							x	x	x
NOx							x	x	x
O3							x	x	x
PM10							x	x	x
RH							x	x	x
SO2							x	x	x
THC							x	x	x
Validate Average Error									**4.24**

Discussion on regularization

1. 其他條件相同時，Regularization 越大，4 個 fold 的 error 看起來越相近。
2. 當 regularization 漸增，在 validation set 上的 average error 有可能會直接變大或者先變小再變大。
3. 理論上，較複雜的 model 在加上 regularization 後應該能降低 variance 造成的 error。但是從 validation set 上來看達到的作用非常有限（error 變小沒多少就變大了、或者直接變大）。可能的原因有：

   (1) Fold 剛好取得很平均，所以 Average error 看不太到 variance 造成的影響，反而 bias 增加造成的影響很明顯。

   (2) 我的 model 都還不夠複雜，是 under-fitting 的情況，應該再增加 model 的複雜程度。

   對於第二點，我一直嘗試增加更多 Feature 進來，卻發現 validation 隨著 feature 的增加越來越差，所以應該已經是 overfitting 了才對。推測這是因為我後來加進來的 feature 都是對接近 target function 沒有幫助的，所以才會雖然 under-fitting 又感覺增加 feature 之後 performance 卻越來越差。

Discussion on learning rate

1. Learning rate 若取得太大很容易就會 overflow，可能是因為 w 和他微分之後的數量級可能有很大差異，尤其是在 w 的 dimension 很大的時候，更是不知道在哪個 dimension 可能會發生問題。
2. 加了 AdaGrad 之後 Learning rate 會隨時間和過去 iteration 的 learning rate 調整，只要選到合適的 initial value，就不容易因為一開始的 learning rate 太大而 overflow。
3. 加了 AdaGrad 之後，Learning rate 會漸漸變小，而且因為新的 learning rate 是由過去的 learning rate 來調整的，而且當 learning rate 很小時，w 的改變量就不會太大，所以 gradient loss 的改變量變小，因此 learning rate 的改變量也在漸漸變小，於是在許多個 iteration 之後 learning rate 的數值就不太會動了，剩下正負號在主導走的方向。