電腦視覺Assignment 3

姓名: 龔鈺閎
學號: 408410046
系級: 資工四

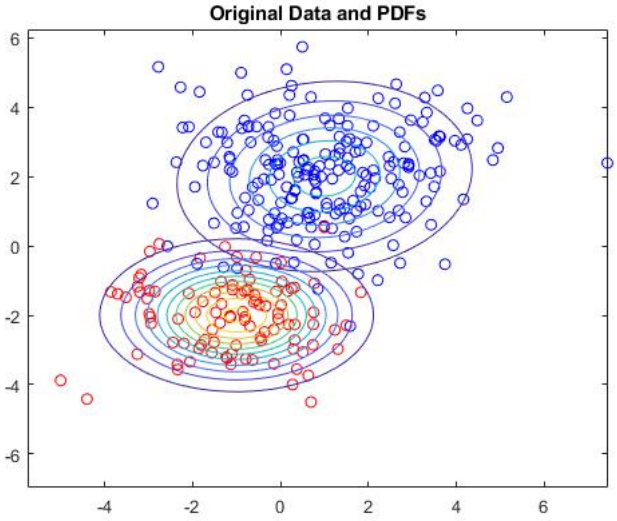
Method Describtion

範例code的解題流程

1. 產生資料、將資料與機率密度函數畫出來
2. 隨機選擇初始的cluster center
3. EM algorithm
4. 計算資料點對應到的每個distribution的機率
5. 畫出資料點以及預測的機率密度函數

1. 產生資料、機率密度函數

範例的GMMExample_2D所產生的資料為2D，所以利用x、y的mean以及covariance matrix可以控制隨機資料點的分布。利用兩組mean以及covariance產生共300筆資料(300 x 2)。機率密度函數pdf是透過在[-6, 6]區間內畫出一個100 * 100個格子，透過格子中的直橫線交叉點的座標下去測試gaussian的機率 並且畫出像是漣漪的圖形。



2. 隨機選擇初始cluster center

首先要先決定要有多少cluster，再隨機挑選對應的cluster center初始值。mu為挑選的初始cluster center ; sigma為每個cluster的variance; phi為每個cluster的機率。

3. EM Algorithm

E step

透過前一步驟初始化所猜的cluster以及variance，算出資料點根據目前所猜想的cluster center所對應的機率。範例假設每個cluster所對應的權重都是平均的，將算出每個cluster對應的機率(pdf)乘上權重(phi)後得到加權後的機率(pdf_w)。
每個cluster都有對應的機率，範例資料總共有兩群，將每個對應cluster的機率(pdf_w())除以所有機率的總和(sum(pdf_w, 2))後得到normalized的機率W。

M step

先將先前假設的cluster center(mu)記起來(prevMu)。根據每個cluster，先利用normalized的機率W算出所有資料點對應到該cluster的機率平均(phi)。接著更新該cluster的mean，也就是cluster center的值，得到新的cluster center(mu)。這裡的想法是資料點對應到cluster center的機率可以解讀成資料對於cluster center更新的貢獻度，就是對應到該cluster的加權機率W。
更新完該cluste center之後，更新cluster的variance，也就是sigma。更新的方法為先將原始資料點減去新的cluster center(mu(j,:))後平方，乘上機率W(i,j)，最後將加總後為sigma_k。最後再除以所有對應到該cluster的機率總和sum(W(:,j))後得到更新的variance，也就是sigma{j}。
更新完每個cluster的mean以及variance之後，就是比對更新前後的mean，如果相同，也就是收斂，就不用再更新了；反之，則反覆EM steps繼續對mean以及variance更新。

EM Algorithm小結

E Step: 透過假設的cluster center，計算每筆資料所對應到每一個cluster的機率。
M Step: 利用E Step算出的機率去更新cluster center以及variance。
重複更新mean、variance，直到mean收斂為止。

4. 計算資料點對應的機率

透過EM Algorithm所更新得出的mean以及variance，就是學習到的GMM，其中mean與variance可以表示GMM中的component。將資料讀入，透過mean以及variance就可以得

到是對應cluster的機率。

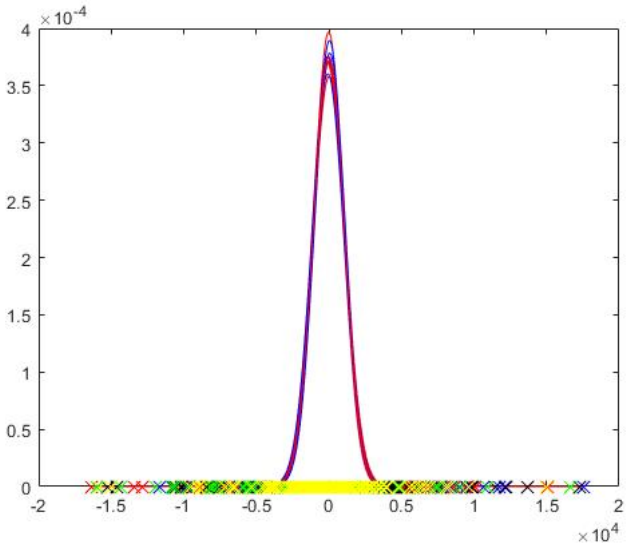
5. 繪出預測機率密度函數

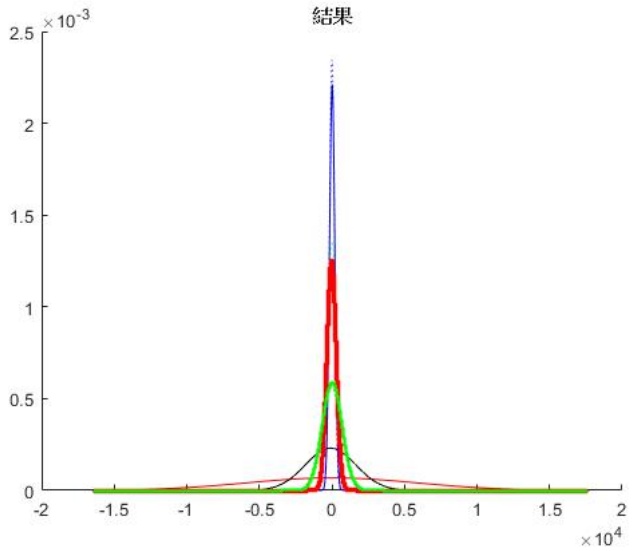
與第一步驟後半部一樣，這邊是用EM Algorithm學到的mean以及variance畫出機率密度函數pdf

Experimental Results

AR Face clustering

根據資料集，總共有10個人的image，每個人分別有13張，每張的維度為1 x 163。直覺的想法是將資料點看成一維的座標，將每張照片的163個點都畫在x軸上，並且透過計算mean以及variance直接算出該人的gaussian component。不過這有個問題，因為人臉大部分都是皮膚色，很多pixel的值都會落在皮膚色的區間，造成每個人的gaussian的mean都會聚集在皮膚色的地方。





跑過幾次 這是其中一次GMM學到的mean

1	2	3	4	5	6	7	8	9	10
-29.1184	0.0149	-61.6185	-100.0473	170.0537	-172.8450	-3.7526	4.8731	-27.9863	29.7302

後來發現gaussian component過於接近，想試看看不同資料維度的做法。

高維度GMM

因為影像一張原本為1 x 163，所以就當作每張影像影163個feature，透過隨機挑10張影像當作cluster center，並且以影像的covariance matrix作為covariance的初始值。透過EM algorithm下去對mean以及covariance matrix更新。

mu

10x163 double

1	2	3	4	5	6	7	8	9	10	11	12	13
-497.8788	-188.0750	-283.2762	-98.9175	100.9966	-259.4851	-386.9650	12.5467	-177.4883	19.6338	-102.8490	-20.2247	-103.5141
504.1427	-335.3187	-382.0775	-164.8921	-34.0404	-230.0811	69.5068	-4.9849	63.1281	98.1888	-131.0822	8.0194	-122.0181
-332.5198	-1.4727	-338.5634	106.4280	74.5871	-249.8516	-56.7297	-77.7074	-60.1971	247.3238	-128.4003	89.8421	-35.7511
-569.4100	-15.9049	-537.9153	-30.7248	-89.7519	-148.2224	-0.3199	-16.6646	-206.7953	-109.2792	-64.5140	63.3930	-233.0963
-788.6165	-415.3650	-522.5014	-199.3696	-114.1923	-45.7674	-39.5357	-66.2354	22.3000	136.7043	-272.4380	-3.2135	-94.6330
391.2839	266.1026	-41.8774	97.0546	-396.5559	-226.5993	-103.8010	-74.0476	-100.5770	135.7773	35.7213	-0.0137	49.0469
-621.8655	-346.2287	-6.4530	-219.7210	93.0550	-310.9942	130.5420	39.1121	-101.3156	246.4090	-103.5110	55.7293	-89.4581
-219.7376	384.1019	-241.9989	-171.6888	200.4214	-304.2006	-159.8243	-140.4934	-147.0247	184.2197	-94.4866	-60.1825	-103.0987
-648.9224	-410.0787	-460.8603	-100.0571	-128.5583	-145.6526	-58.7019	83.6362	25.0878	159.1888	-103.2791	-63.0069	-113.5423
-297.1866	-199.0131	-343.4640	-253.8590	117.4838	-346.5716	139.8700	-121.2677	-33.0118	97.1825	-122.0649	25.8644	-32.5884

更新過後的10個cluster center的mean。

130x10 double

1	2	3	4	5	6	7	8	9	10
0.2201	0.0514	0.1001	0.0210	0.1586	0.0780	0.1864	0.0090	0.0927	0.0827
0.0146	0.1188	0.1607	0.0878	0.0885	0.0605	0.1640	0.1279	0.0724	0.1050
0.2012	0.1137	0.1207	0.0862	0.0814	0.0574	0.0820	0.2001	0.0494	0.0080
0.0809	0.0019	0.1528	0.0389	0.1440	0.0438	0.0892	0.2472	0.0997	0.1016
0.0771	0.0529	0.0359	0.0732	0.2194	0.1607	0.0470	0.2013	0.1308	0.0017
0.1041	0.0434	0.1619	0.0763	0.0879	0.1784	0.1353	0.1918	0.0071	0.0138
0.1238	0.0210	0.0857	0.1090	0.0169	0.1368	0.0959	0.2298	0.0670	0.1140
0.1762	0.0768	0.0058	0.1057	0.2777	0.0237	0.0994	0.1897	0.0205	0.0245
0.0878	0.1164	0.1631	0.0554	0.1781	0.1824	0.0873	0.0291	0.0781	0.0223
0.0120	0.0547	0.0014	0.0356	0.1929	0.1907	0.0551	0.2918	0.0933	0.0725
0.1794	0.1264	0.1539	0.0360	0.0749	0.0126	0.2099	0.0302	0.1243	0.0524
0.0617	0.0492	0.0719	0.0696	0.1814	0.1451	0.2500	0.0269	0.0957	0.0485
0.1942	0.0186	0.1300	0.1699	0.1755	0.0827	0.0752	0.0160	0.0348	0.1031
0.0831	0.1229	0.0391	0.1473	0.1932	0.0607	0.0960	0.1417	0.0538	0.0621
0.1245	0.0279	0.0590	0.1160	0.1584	0.2058	0.1226	0.0154	0.1425	0.0279
0.2004	0.1233	0.1394	0.0158	0.0039	0.1150	0.0122	0.2022	0.0772	0.1107

test data每筆對應到cluster的機率

準確率

163 * 1	1 * 163
12.3%	12.31%
10%	9.23%
10.77%	12.31%
14.62%	9.23%
13.08%	9.23%

一維的準確率是透過163個pixel中統計預測最高的那一個cluster。
1 x 163的準確率是直接用1 x 163的向量下去算對應cluster的機率。
五次163 features的GMM都跑了1000次的iteration。都沒有在1000次的iteration中收斂。

Discussion of results

跑過幾次一維的作法，出來的準確率最高為14.62%。因為一維的資料過於集中，不像範例那樣mean有明顯的差異而且資料點都集中在特定區間。
跑過163 features的GMM，發現準確率沒有比一維的來的好。推測的原因是因為再計算的時候，pdf會出現0的情況，造成再計算的時候divided by 0的情況。為了避免這種情況發生，我有逐一trace了可能會發生的情況，並且都加上了一點小小的noise。但這些noise可能會對結果造成影響。
能夠提升準確率的方法，對於資料的前處理，可能可以透過gaussian blur將資料一部分的noise清除之後，保留重要的feature，再進行訓練。
至於mean收斂的部分，理論上在跑EM的時候，應該會收斂，但是目前實驗的結果是沒有。推測是在計算過程中加上的noise會影響到實驗的結果，但資料本身也會對於準確率有所影響。如果資料的covariance matrix是 well scaled的話，就不須透過增加noise去避免sigular covariance matrix的問題了，對於GMM的訓練與收斂也會影響。
準確率的部分，我認為提高training data的資料量以及iteration的次數，對於準確率會有正向的幫助。有觀察到在跑EM的時候，跑了1000次也沒有辦法收斂，提高iteration的

次數也許是一種方法。但總資料量應該才是左右訓練效果的主因。

Problem and difficulties

1. 在算每個cluster的pdf的時候，因為資料維度太大(162 x 162)所以determinant算出來會是無限大，導致在更新的時候會有問題。

這部分當初只想到determinant值過大，沒有考慮到covariance matrix是不是singular，也就是在計算gaussian的時候沒有辦法求出inverse of covariance matrix。解決的方法，看到網路上的資料，都是在更新covariance matrix的時候在對角線上加上一個小小的noise。

```
% Divide by the sum of weights.
sigma{j} = sigma_k ./ (sum(W(:, j))) + diag(rand(1, 163) * exp(-6));
```

在更新covariance的後面加上一個三角矩陣，對角線上的值就是隨機產生的noise。

2. 在算PDF的時候，會出現機率為NaN，導致運算錯誤

我在執行的時候有觀察到在計算每個cluster的pdf，機率會出現0，讓後面的weighted probablity會出現 divided by 0的情況。為了避免除以0，我也是參考了網路上的方法，就是在計算pdf的時候加上小小的noise。

3. 若知道分類 可以在train的時候就先算出不同cluster的mean以及variance，這樣在測試的時候，直接下去算

如果訓練資料夠多的話，其實就跟範例給的一樣，可以先透過計算mean，covariance。有了這些資料，就好像學到了cluster center以及variance。將測試資料讀入就能給出相對應的機率。

4. 算出來的mean與label的關聯

算出來的mean，要怎麼跟training data的label做對應呢?原本想說直接用training data的一部分下去算機率，但發現GMM並沒有想像中的準確，會出現重複的問題。後來想到可以先算每個label的cluster center，再透過歐式距離去判斷預測的mean與哪一個label的center最接近。不過，還是一樣的問題，GMM的準確率不夠高。

後來想到一個方法是在mean的initial guess，就先挑選label 1 ~ 10的資料點當作初始，不像範例一樣隨機挑選。這樣做的話，可以確保在更新mean的時候，都是保持label 1 ~ 10的順序。

```
indeces = randperm(13);
mu = [];
for i = 1:10
   mu = [mu; X((i - 1) * 13 + indeces(i),:)];
end
```

Reference

1. singular covariance matrix in GMM: **https://stats.stackexchange.com/questions/219302/singularity-issues-in-gaussian-mixture-model**

   (https://stats.stackexchange.com/questions/219302/singularity-issues-in-gaussian-mixture-model)