MDA_HW3_KMeans

107062321 王劭元

在這次的作業中我主要共用到了 6 個 mapper，2 個 reducer

mapperE_dist：在這個 mapper 中，我是將每個點去與 centroid 算出他們之間的 Euclidean distance 後，以我希望的形式包好後回傳

mapperE_min：在這個 mapper 中，傳進來的資料會是 data 中的點，後面接著十個 centroid，我用迴圈掃過 10 個 centroid 後，會得到是哪個 centroid 距離最短，並且距離為多少，最後要回傳時，我除了回傳是哪個 centroid 之外，我還會回傳這個 centroid 的 index 以利於後面可以 sortByKey，讓最後 csv 檔的輸出可以順利一點。並且我在回傳時，我在最後的 KV pair 就會直接回傳這個點所產生的 cost 為多少，這樣最後我對 data 所有點的這個值用 api sum()起來，就可以得到這次的總 cost 為多少。除此之外，由於發現 data 中有重複出現的點會影響到 cost，所以我還會去算若同樣的點後面還接了幾個，最後的總 cost 會去乘上點的數量。

mapperM_dist：在這個 mapper 中，我是將每個點去與 centroid 算出他們之間的 Manhattan distance 後，以我希望的形式包好後回傳

mapperM_min：在這個 mapper 中，與 mapperE_min 實作的方式幾乎一樣，我

只是為了要回傳不同的 cost。所以這邊只有在回傳的 cost 算法不一樣，其他都跟 mapperE_min 一樣。

mapperNew_cen：在這個 mapper 中，傳進來的資料會是其中一個 centroid 接上，現在屬於這個 centroid，或者說新的這個 cluster 的點。所以要計算新的 cluster 的中心時，就把後面用迴圈全部掃過，並且把各 dimension 的值加起來平均後，即可得到新的 centroid。

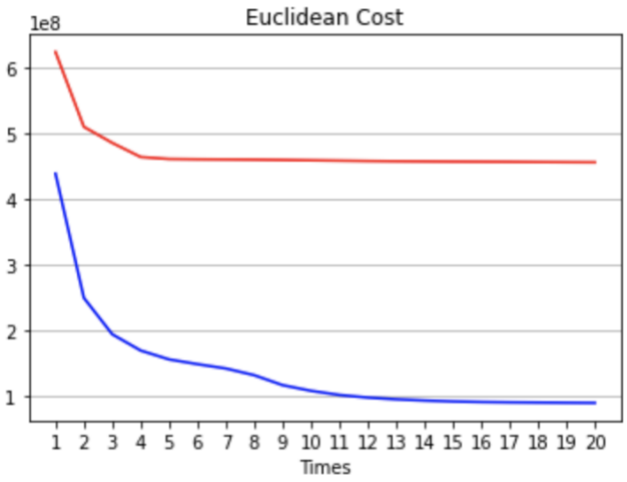
mapperinit：在這個 mapper 中，我為了讓每次 iteration 完，紀錄 centroid 的 rdd 回復成我一開始的型態所用的。會回傳 (0, centroid)。

Reducer：兩個 reducer 的功用都蠻像的，都是為了讓我可以將資料串接在一起。一次是在每次 iteration 一開始，我的 centroid 和 data 這兩個 rdd 的 kv pair 都長成 (0, centroid) (0, data)，的形式，所以我先把他們 join 起來後，算完每一個點和 centroid 的距離後，為了要去計算 min distance，要把同一個點對不同的 centroid 的這些 pair 串起來，故我這裡用了一次 reduceByKey(lambda a, b:a+b)。第二次是在把每個 centroid 的與某個 data point 的 cost 算完後，為了要算出總 cost 要把所有屬於這個 cluster 得 data point 串在 centroid 的後面，於是我在這邊用了第二次的 reduceByKey(lambda a, b:a+b)。

(a)

(1) A plot of cost vs. iteration for 2 initialization strategies

	C1	C2
0	6.236603e+08	4.387478e+08
1	5.098629e+08	2.498039e+08
2	4.854807e+08	1.944948e+08
3	4.639970e+08	1.698048e+08
4	4.609693e+08	1.562957e+08
5	4.605378e+08	1.490942e+08
6	4.603131e+08	1.425085e+08
7	4.600035e+08	1.323039e+08
8	4.595705e+08	1.171710e+08
9	4.590211e+08	1.085474e+08
10	4.584907e+08	1.022372e+08
11	4.579442e+08	9.827802e+07
12	4.575580e+08	9.563023e+07
13	4.572901e+08	9.379331e+07
14	4.570506e+08	9.237713e+07
15	4.568922e+08	9.154161e+07
16	4.567036e+08	9.104557e+07
17	4.564042e+08	9.075224e+07
18	4.561778e+08	9.047017e+07
19	4.559869e+08	9.021642e+07



(b) Percentage improvement values and your explanation

```
Change with c1 26.885383292517258
Change with c2 79.437750291599
```

由於 Euclidean 計算 cost 的方式為距離的平方，所以對於 cluster 距離遠近、離群值較為敏感，故在這種情況，使用 c2 事先刻意挑好讓最遠的點可以自己形成一種 cluster 的方式，最後得到結果以及下降的幅度也較好。

(c)

(1) Euclidean distance for all pairs of centroids, with c1 started

Euclidean	1	2	3	4	5	6	7	8	9	10
	0	692.157887	3490.25864	205.750279	346.718823	512.612247	444.731001	566.201992	1282.77084	307.669128
		0	2798.80105	897.658986	1038.82689	1204.0782	1136.32734	1257.44953	669.890228	412.076077
			0	3695.11419	3836.90664	4002.68908	3934.87156	4056.13557	2294.57964	3195.9239
				0	142.438874	309.506324	241.730115	363.262895	1474.94542	504.634116
					0	167.1498	99.5455433	220.901784	1615.85235	646.930564
						0	67.9118611	53.7898912	1782.20305	814.07615
							0	121.63372	1715.2532	746.335559
								0	1835.63967	867.823079
									0	975.320423
										0

(2) Manhattan distance for all pairs of centroids, with c1 started

A	B	C	D	E	F	G	H	I	J	K
Manhattan	1	2	3	4	5	6	7	8	9	10
	0	728.924314	3797.89908	212.18109	374.890422	577.402076	499.157894	645.769777	1731.06431	406.701225
		0	3072.88869	935.885338	1100.83309	1303.89572	1225.35171	1372.09221	1005.29305	490.928058
			0	4001.03805	4170.30453	4372.78872	4294.95283	4440.71977	2513.42266	3396.42
				0	171.365154	375.247921	296.254724	443.498445	1934.08696	609.749322
					0	204.522924	125.596786	272.934913	2102.86492	779.397227
						0	79.4016844	69.5898763	2306.38025	983.019681
							0	147.865709	2227.55586	904.37025
								0	2374.54543	1050.91622
									0	1327.58398
										0

(3) Euclidean distance for all pairs of centroids, with c2 started

Euclidean	1	2	3	4	5	6	7	8	9	10
	0	15760.1225	14110.8344	9045.32023	5567.68452	1924.62408	1100.85905	402.89055	2105.44258	3169.00377
		0	11524.5057	6743.8841	10192.525	14455.1194	14682.451	15362.418	13674.7075	12597.0396
			0	9545.8794	10883.3822	12233.9598	13208.0029	13786.4842	12508.9574	11938.3761
				0	3494.22242	7718.22201	7957.77595	8644.80704	6947.82064	5876.3302
					0	4404.56259	4492.45821	5169.93729	3488.15852	2407.91879
						0	1182.86419	1615.78824	1313.32749	2153.77147
							0	698.488136	1010.19767	2085.46068
								0	1702.79266	2768.60772
									0	1080.53494
										0

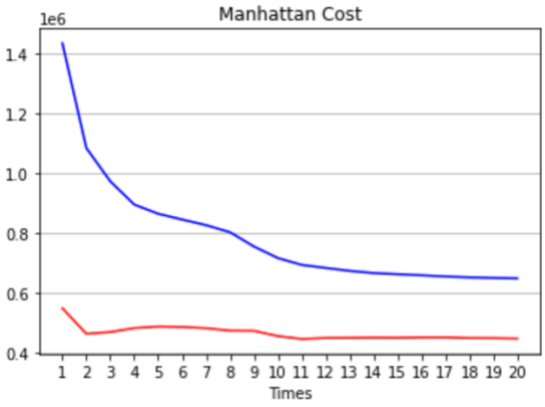
(4) Manhattan distance for all pairs of centroids, with c2 started

Manhattan	1	2	3	4	5	6	7	8	9	10
	0	15772.6149	20215.646	9533.17085	5604.20049	3088.05432	1311.03916	471.26572	2369.41216	3349.65709
		0	16003.499	7219.19667	10221.031	16105.3475	14909.1695	15434.46	13950.576	12776.8831
			0	10690.4843	14613.552	17509.9028	18912.6054	19748.9357	17851.8068	16873.2437
				0	3935.29267	8896.38921	8228.35508	9065.40433	7168.73296	6190.67931
					0	5893.07013	4696.97538	5221.25281	3737.707	2564.17054
						0	1781.82267	2619.81139	2162.80215	3337.74626
							0	840.722524	1068.93997	2137.78826
								0	1901.20876	2883.73454
									0	1176.45043
										0

(b)

(1) A plot of cost vs. iteration for 2 initialization strategies

	C1	C2
0	550117.142000	1.433739e+06
1	464869.275879	1.084489e+06
2	470897.382277	9.734317e+05
3	483914.409173	8.959346e+05
4	489216.071003	8.651283e+05
5	487629.668550	8.458466e+05
6	483711.923214	8.272196e+05
7	475330.773493	8.035903e+05
8	474871.238846	7.560395e+05
9	457232.920115	7.173329e+05
10	447494.386197	6.945879e+05
11	450915.012577	6.844445e+05
12	451250.367073	6.745747e+05
13	451974.595540	6.674095e+05
14	451570.364070	6.635566e+05
15	452739.011366	6.601628e+05
16	453082.730287	6.560413e+05
17	450583.670860	6.530368e+05
18	450368.749317	6.511124e+05
19	449011.363726	6.496890e+05



(b) Percentage improvement values and your explanation

```
Change with c1:  18.378954327236887
Change with c2:  54.685694348134085
```

由於 Manhattan 在計算 cost 的方式相對來說比較不敏感，Manhattan 更希望你能將集中很多點的大群即使距離很近，也可以分成不同 cluster，故在這種情況，一開始還特地將距離拉遠來看，最後達到的效果沒有比較好。

(c)

(1) Euclidean distance for all pairs of centroids, with c1 started

Euclidean	1	2	3	4	5	6	7	8	9	10
	0	2219.17728	9948.04408	528.699758	413.365061	827.718886	681.03499	917.127383	832.147434	729.056349
		0	7767.9456	2734.04985	2628.49081	3044.47787	2898.71289	3133.46013	1812.45457	1491.35735
			0	10433.0614	10361.3675	10773.5308	10626.4886	10862.9658	9340.27523	9236.84002
				0	221.372794	375.156188	249.379188	457.259653	1156.58338	1251.15835
					0	415.989985	270.748792	505.071067	1171.96421	1137.13527
						0	147.046974	89.4909166	1529.46401	1553.12381
							0	236.514622	1391.55042	1407.4044
								0	1613.55579	1642.12869
									0	709.407786
										0

(2) Manhattan distance for all pairs of centroids, with c1 started

Manhattan	1	2	3	4	5	6	7	8	9	10
	0	2341.01722	11929.3002	651.187488	496.331521	947.743236	770.737383	1056.7995	1260.51056	737.713573
		0	9597.44119	2778.94576	2830.14453	3280.35917	3104.28577	3388.98265	2380.46096	1605.27013
			0	12323.2876	12421.2631	12871.4834	12695.5542	12979.1332	10775.9392	11196.787
				0	335.951213	558.469258	382.46333	667.53323	1653.82589	1379.16517
					0	452.861331	276.326491	561.849249	1755.10553	1226.66035
						0	177.593162	110.217624	2205.30738	1677.66686
							0	287.429708	2028.90162	1500.99341
								0	2314.66745	1786.81132
									0	1006.36783
										0

(3) Euclidean distance for all pairs of centroids, with c2 started

Euclidean	1	2	3	4	5	6	7	8	9	10
	0	15747.2342	14100.1447	9032.33302	5554.78669	2006.70267	1338.16113	514.627038	1571.24342	3022.66088
		0	11524.5057	6743.8841	10192.525	14474.5541	14412.0566	15239.8771	14328.2262	12731.3976
			0	9545.8794	10883.3822	12167.7939	13125.351	13684.6068	12643.9856	12006.3946
				0	3494.22242	7742.62812	7694.2767	8521.19786	7588.40454	6009.82022
					0	4452.97168	4219.76057	5047.51626	4167.63653	2542.56935
						0	1405.10908	1637.72944	910.994388	2124.26336
							0	827.840658	566.551017	1684.51601
								0	1081.37933	2511.45886
									0	1649.38917
										0

(4) Manhattan distance for all pairs of centroids, with c2 started

Manhattan	1	2	3	4	5	6	7	8	9	10
	0	15757.6913	20200.2594	9517.66823	5588.85363	3281.48825	1430.20868	602.954849	2102.55398	3211.45576
		0	16003.499	7219.19667	10221.031	16325.2705	14506.4859	15335.9574	14980.0561	12922.9314
			0	10690.4843	14613.552	17521.5177	18775.1215	19602.2628	18111.8854	16995.1335
				0	3935.29267	9116.0245	8090.51019	8918.81312	7771.22208	6312.53001
					0	6110.8325	4293.5019	5123.06681	4768.923	2710.0565
						0	1855.57991	2682.56923	1358.79589	3413.03618
							0	833.430282	674.82757	1784.51205
								0	1500.82488	2613.99731
									0	2062.25107
										0