ML HW8 report

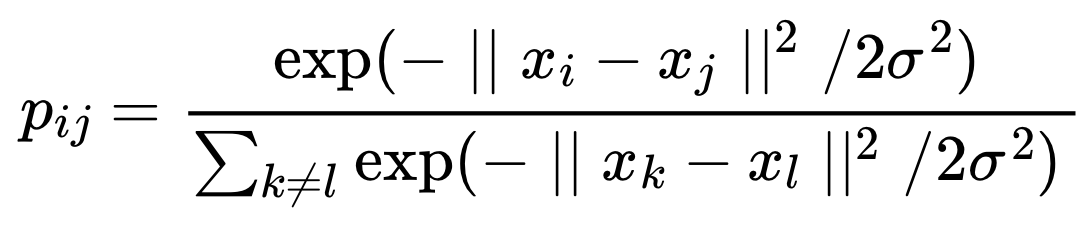
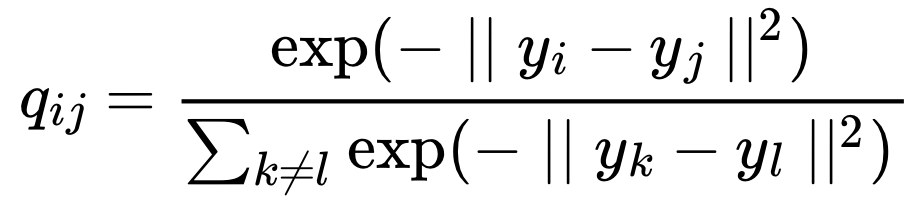
資科工碩一

0756138

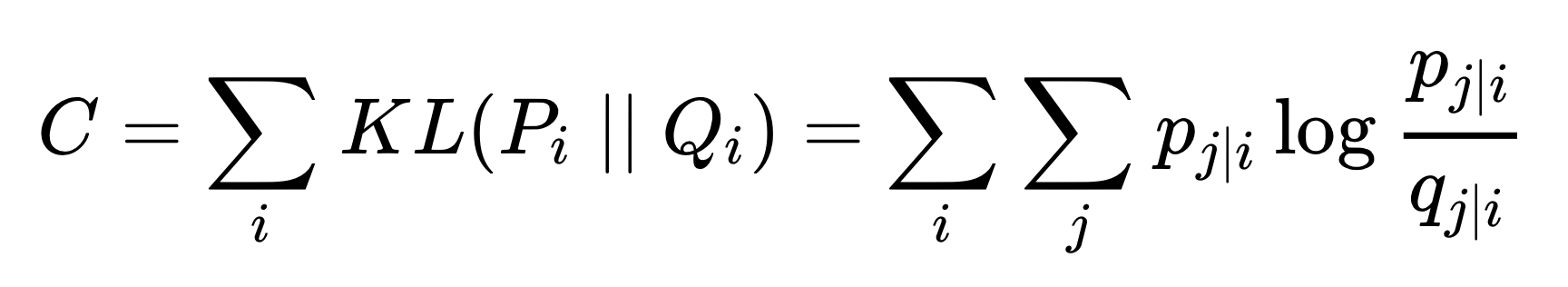
黃伯凱

**Symmetric SNE**

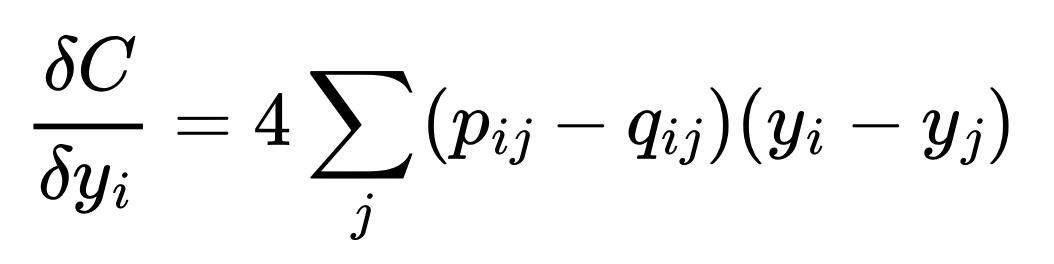
一般來說，將data分成多個class的時候會利用data之間的距離來區分，而SNE則是將距離轉換成機率來表達點跟點之間的similarity。P代表高維時的similarity，Q則代表低維時的similarity，使用的gaussian來計算similarity。

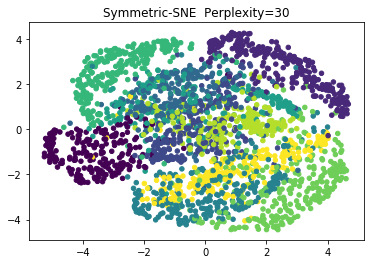
 

目標函數則是兩個維度之間的差距，由於上述是計算出各自的分佈，所以差距就要用KL divergence來計算。



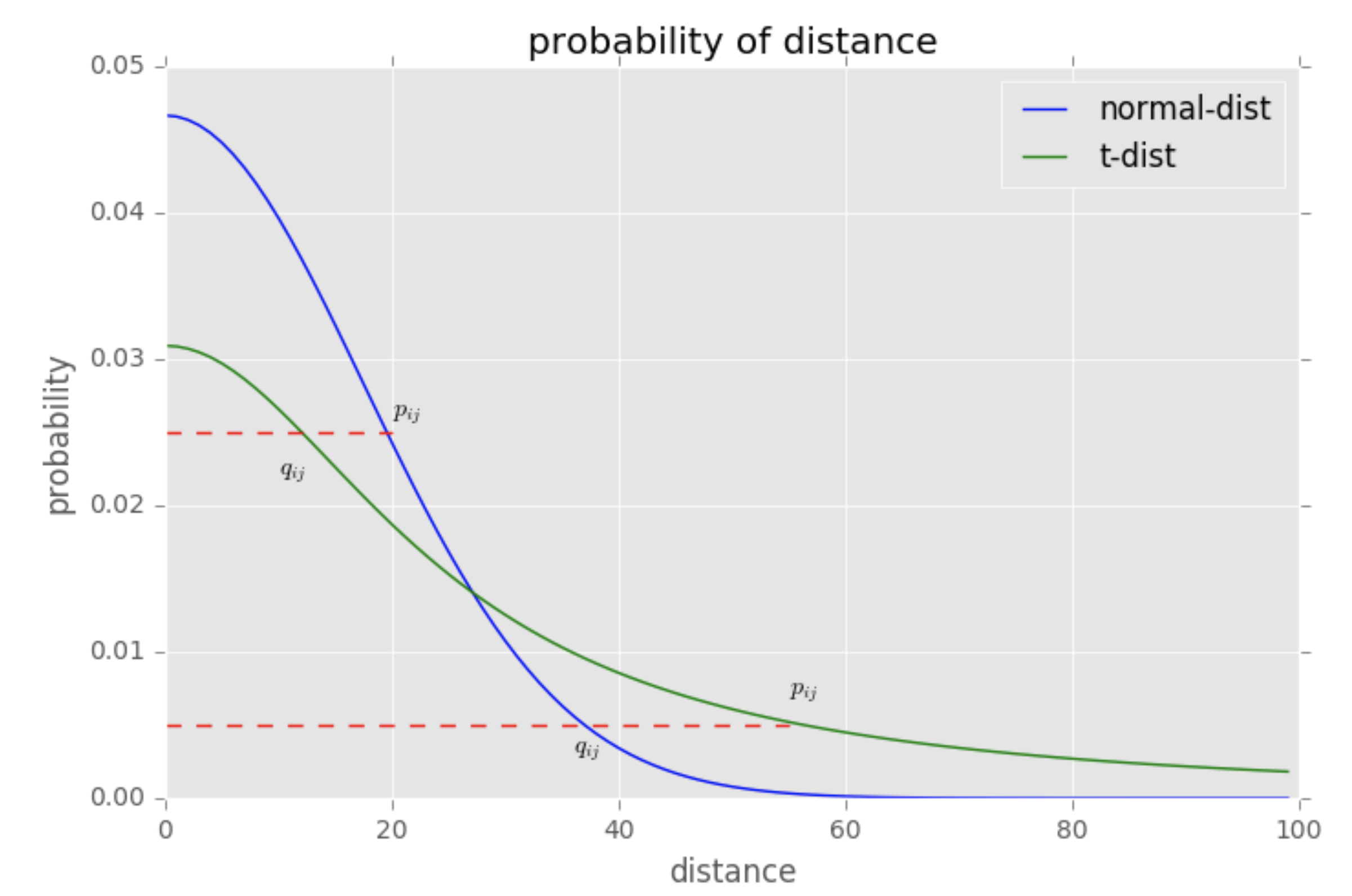
Gradient的算法則如下：





**T – SNE**

如Symmetric-SNE的結果，即使能夠成功分辨出不同的class，在視覺化上我們卻難以把不同class的data明顯區分開來。所以t – SNE改成使用t – distribution來計算similarity：

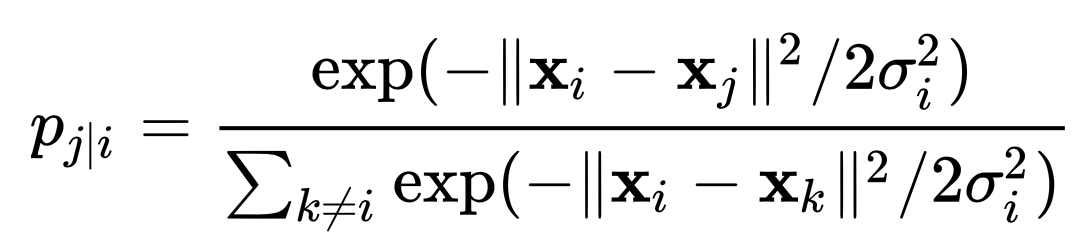
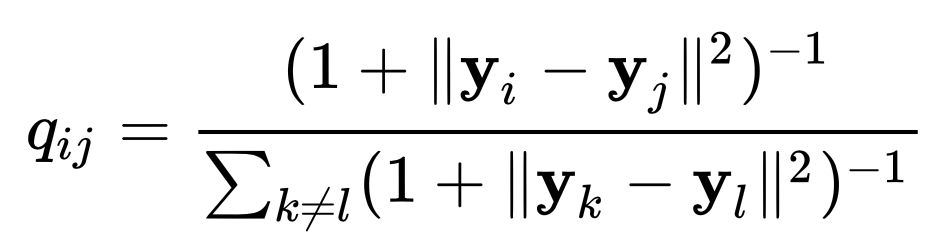


如上圖所示，藍色的線是Symmetric – SNE使用的Gaussian，綠色的線是t – SNE使用的t – distribution。可以得到兩個差異：

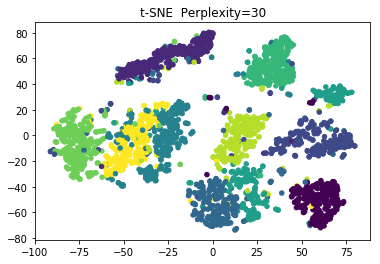
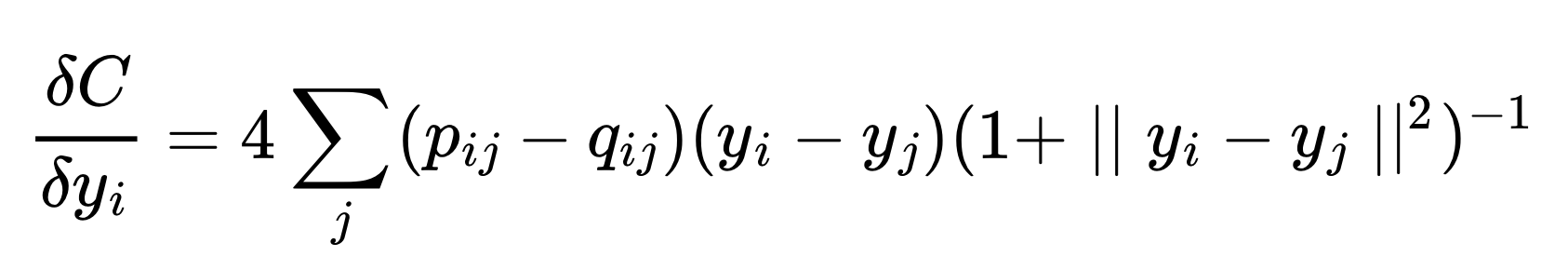
1. 藍線轉換成距離之後，幾乎都是在60以內；綠線轉換成距離之後，距離甚至可以超過100。從此現象可以知道t – distribution可以把data之間的距離拉開。
2. t – distribution在短距內的data數量並沒有像Gaussian那樣那麼多，所以在gaussian低維靠很近的data，t – distribution可以有效把那們分開。

根據以上現象，可以確定t – distribution可以將不同class的data在低維時確實地分開。以上圖兩個紅色虛線為例，p=0.025時t – distribution得到的距離 < Gaussian；p=0.005時t – distribution得到的距離 > Gaussian。再次驗證t – distribution可以把近距data拉更近，遠距data拉更遠。

由於利用t – distribution來算similarity，P和Q變成如下：

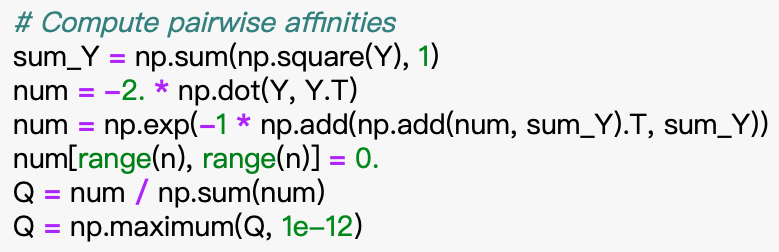
Gradient也要跟著改變：



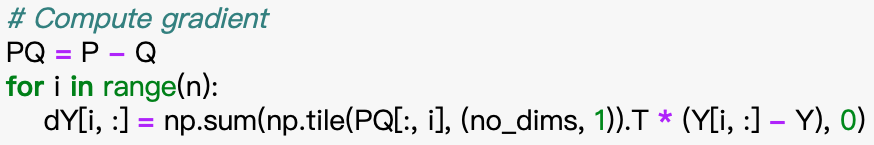
**Code Difference**

**Symmetric – SNE**

根據上面的式子，要改Q的算式，所以要改這邊的num

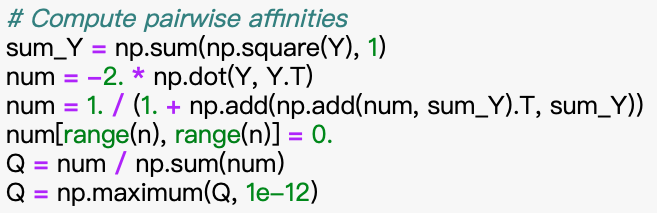


Gradient則要從這邊改

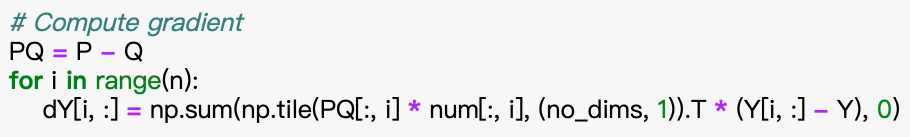


**T – SNE**

根據上面的式子，要改Q的算式，所以要改這邊的num

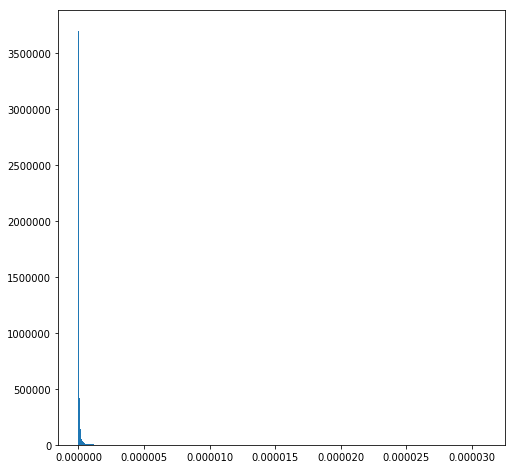
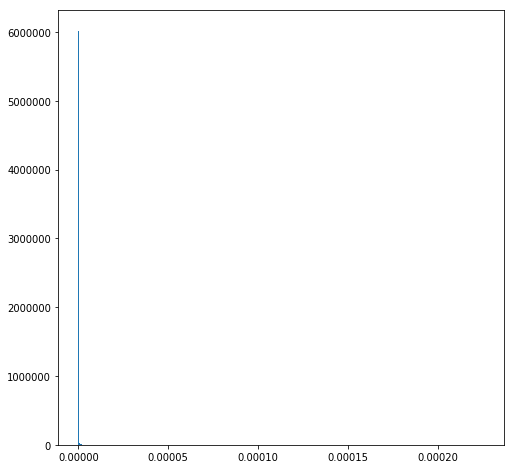


Gradient則要從這邊改



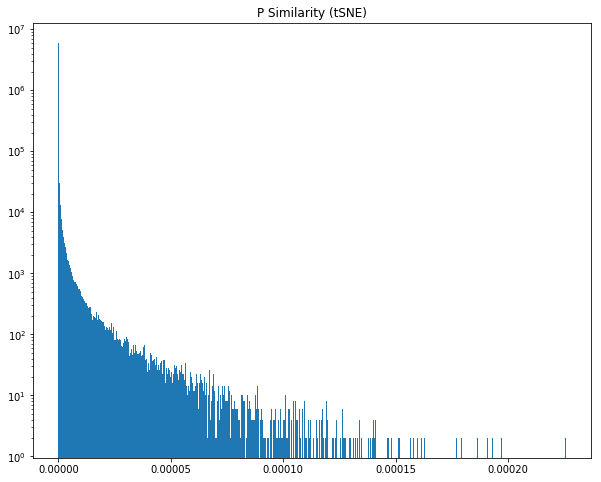
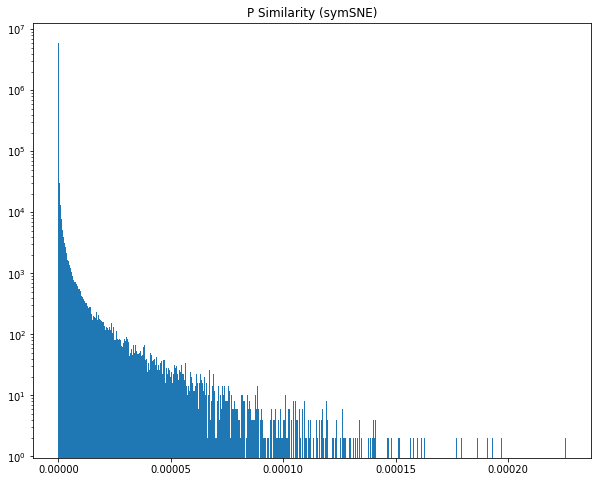
**Distribution of Pairwise Similarity**

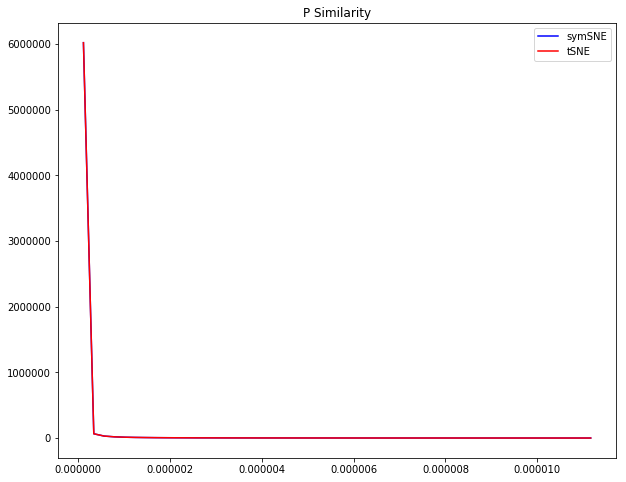
由於大部分的距離都非常小，會長的像下面這種分布：



所以以下的分佈都有先取log再畫出來

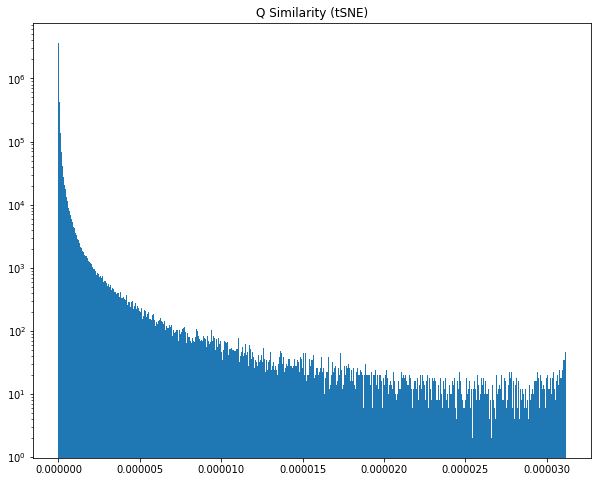
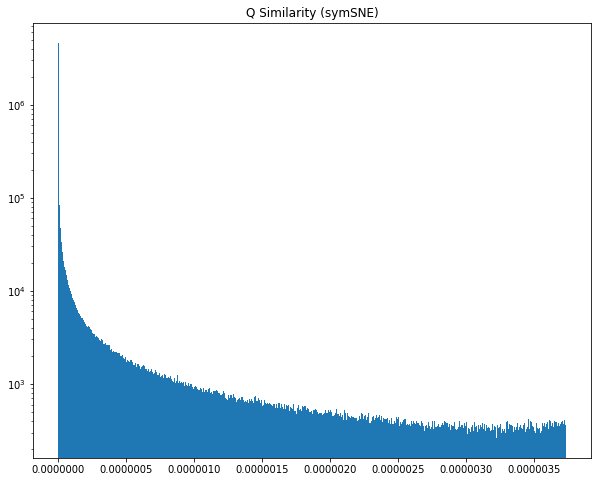
**P Similarity**

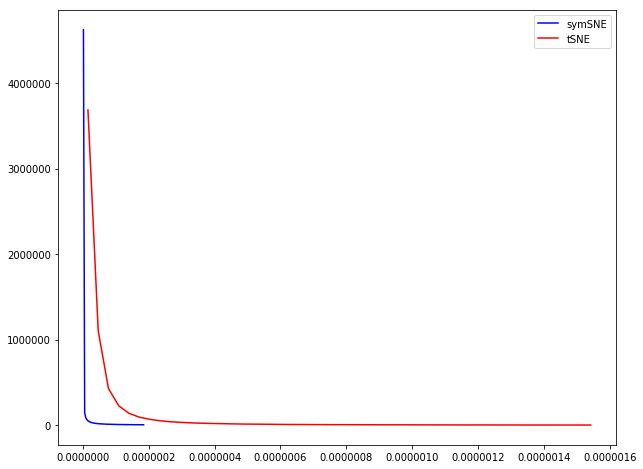




因為P Similarity的計算方式沒有改變，所以兩種方法的P Similarity會長一模一樣。

**Q Similarity**



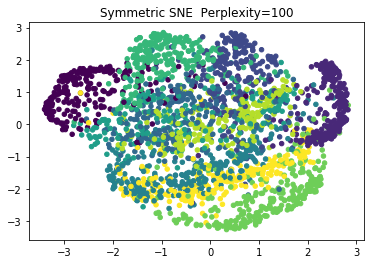
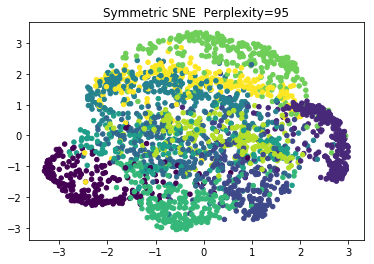
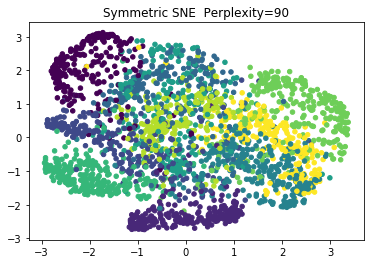
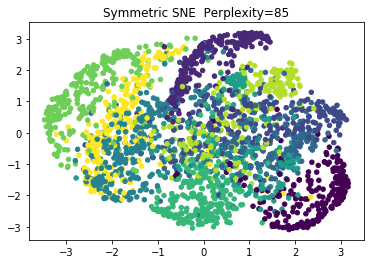
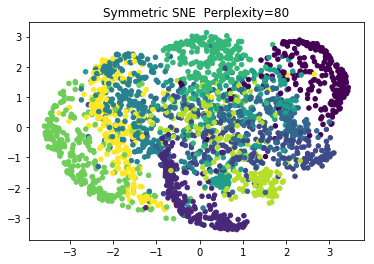
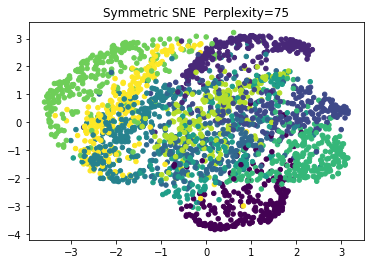
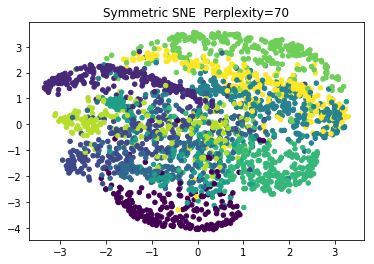
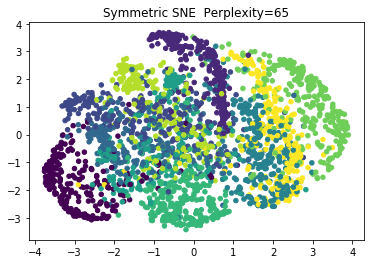
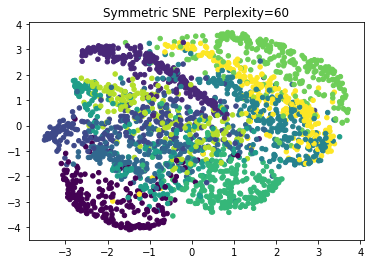
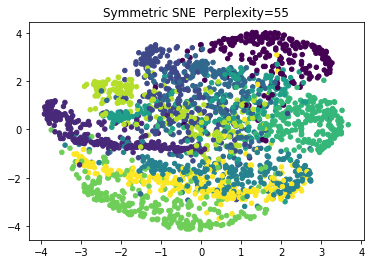
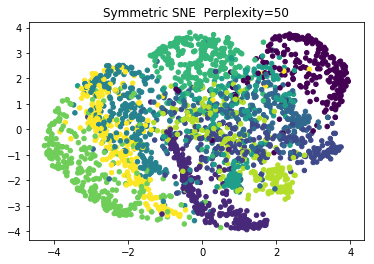
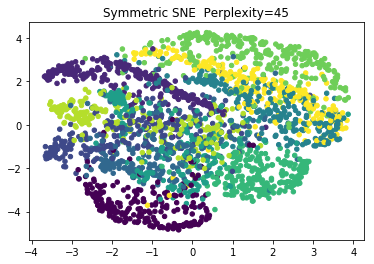
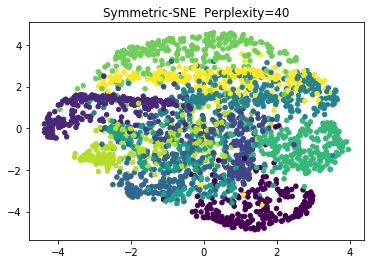
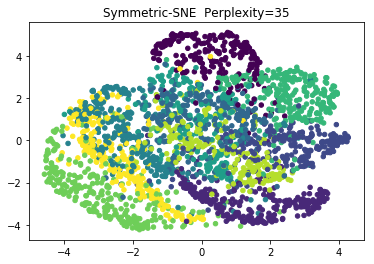
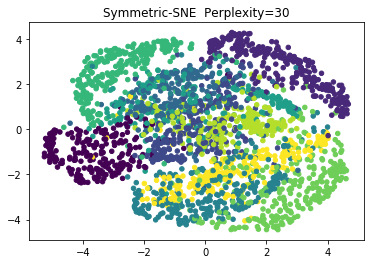
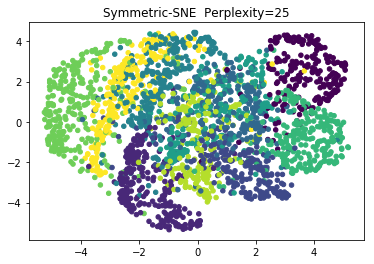
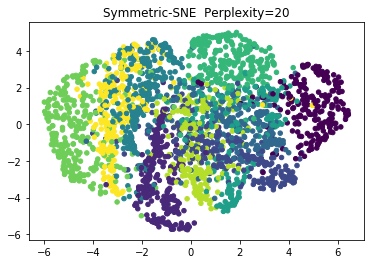
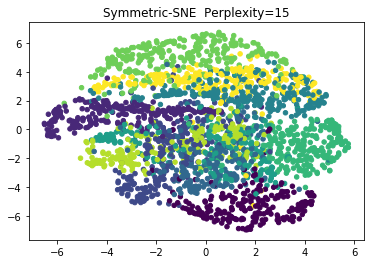
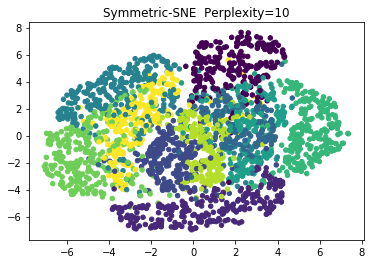
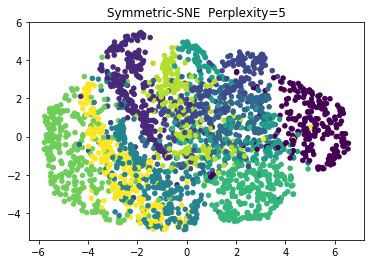


(x軸為Similarity，y軸為data數量)

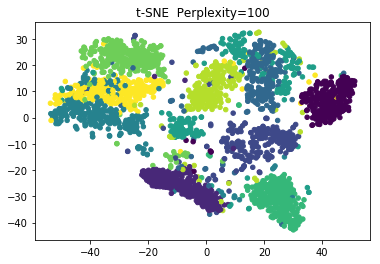
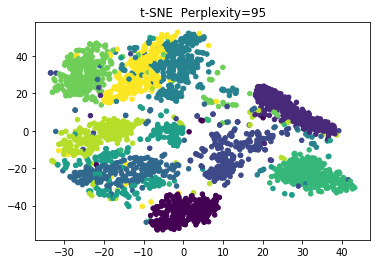
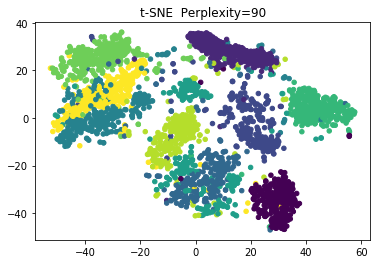
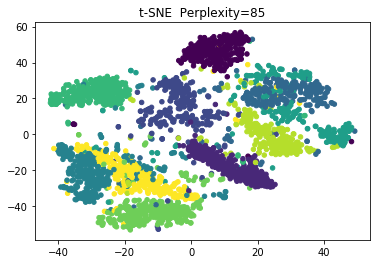
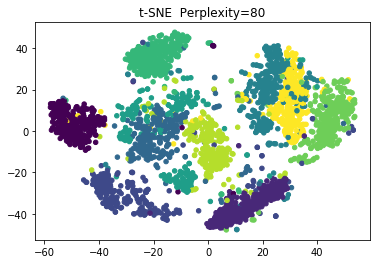
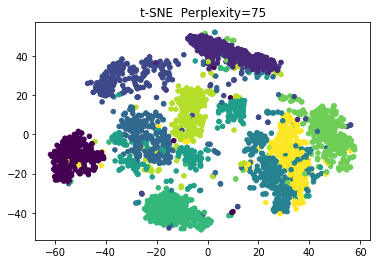
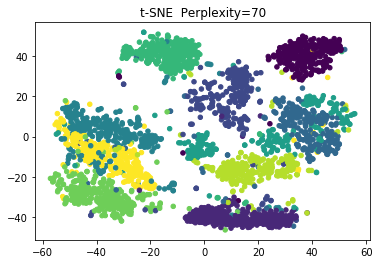
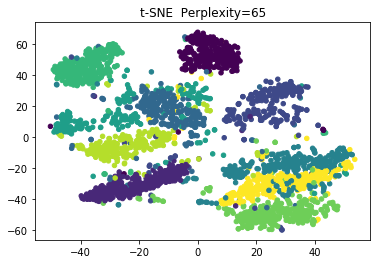
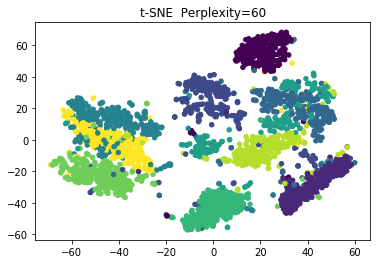
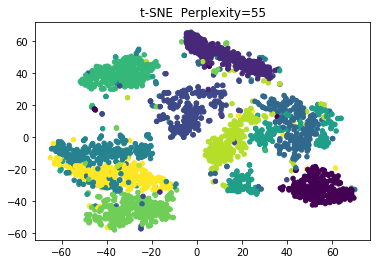
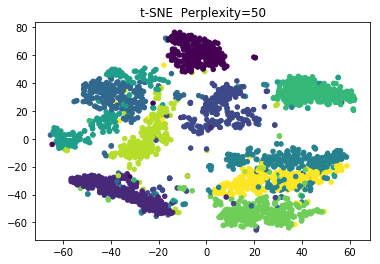
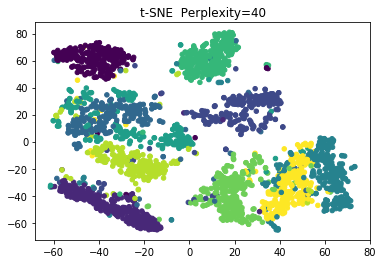
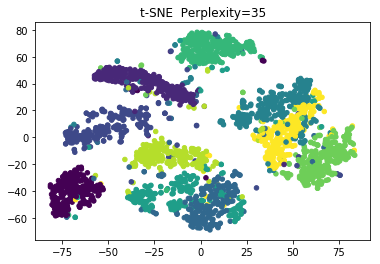
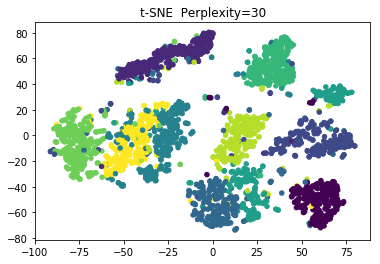
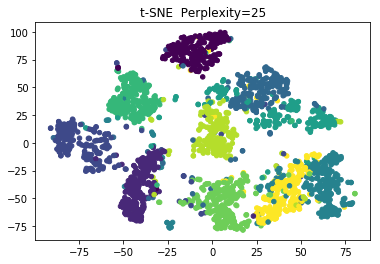
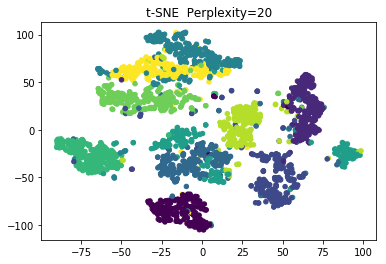
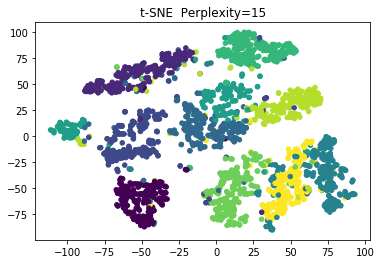
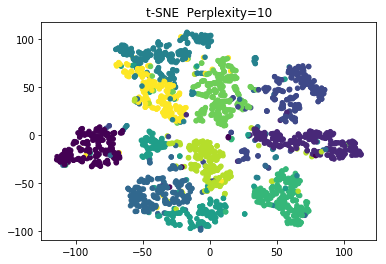
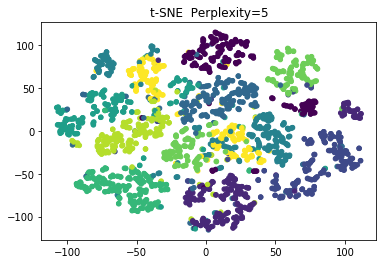
可以發現Symmetric – SNE的結果大多距離非常遠，因為大部分的data Similarity都非常接近0。反觀t – SNE，Similarity為0.0000002的data數量其實還是蠻多的，代表距離近的data比Symmetric – SNE還多。

**Different Perplexity**

**Symmetric – SNE**

****

**T - SNE**



Symmetric – SNE的結果不論perplexity調多大或多小基本上不太會有什麼差異，因為Symmetric – SNE本來就不會把每個class之間的距離拉大，所以perplexity從5~100看起來都沒什麼差異。

而t – SNE就可以蠻明顯的看得出來大概在perplexity為15~40時可以把每個class切割得很清楚，perplexity太大或者太小的時候不同class混在一起的機率就比較高。但是perplexity在5~100的區間裡時，t – SNE的結果確實是比Symmetric – SNE好的。