T-distribution Stochastic Neighbourhood Embeding

t-SNE is one of the other methods for Dimensionality Reduction . It works wonder when it comes to visulaization of high dimensional datas. t-SNE tries to balance attention between the local and global aspects of the data. The two main parameters of t-SNE are:

a.) Perplexity : It gives a sense about number of close close neighbours each point(Datapoint) has.

b.) No. of Iterations : Until and unless your visualization comes to a stable point the iterations in t-SNE should be carried on which is controlled using this parameter.

Although the results we get through t-SNE are very impressive there could be certain misreadings through visulization point of view which are as follows :

a.) Visulaization patterns will keep on changing with change in perplexity and number od iterations . Also with the same perplexity and number of iterations if you run the code again you may end up getting different visualization pattern.

b.) Cluster sizes in a t-SNE plot means nothing, since t-SNE has this tendency to expand the cluster which has smaller size and to shrink up the larger clusters

c.) Distance between clusters might not be same in t-SNE plots as it would be in the original data

d.) Random noise doesn't always look random in t-SNE plots

In [2]:

```
import pandas as pd
import seaborn as sns
import numpy as np

data = pd.read_csv('./Data/MNISTtrain.csv')
```

In [3]:

```
data.head(5)
```

Out[3]:

	label	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	...	pixel774	pixel775	pixel776	pixel777	pixel778	pixel77
0	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
2	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
3	4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	

5 rows × 785 columns

In [4]:

```
label = data['label']
dataset = data.drop('label' , axis = 1 )
```

In [5]:

```
label.value_counts()
```

Out[5]:

```
1    4684
7    4401
3    4351
9    4188
2    4177
6    4137
```

```
0    4132
4    4072
8    4063
5    3795
Name: label, dtype: int64
```

In [6]:

```
dataset.shape
```

Out[6]:

```
(42000, 784)
```

Visualization of the Hand-written Data

In [7]:

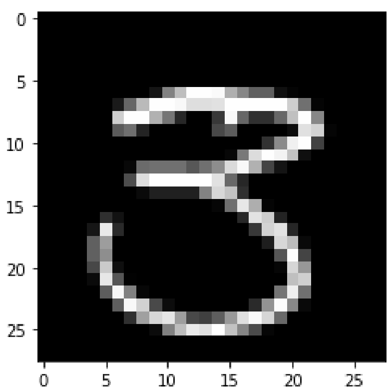
```
import matplotlib.pyplot as plt
idx = 7

pix = dataset.iloc[idx].as_matrix().reshape(28,28)
plt.imshow(pix , cmap = 'gray')

print (label[idx])
```

```
C:\Users\Nrohlable\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: FutureWarning: Method .as_
matrix will be removed in a future version. Use .values instead.
  after removing the cwd from sys.path.
```

```
3
```



Process for T-SNE :

In [8]:

```
from sklearn.manifold import TSNE
from sklearn.preprocessing import StandardScaler

standarddata = StandardScaler().fit_transform(dataset)
print ('Shape of Standaradized Data', standarddata.shape)
```

```
Shape of Standaradized Data (42000, 784)
```

In [33]:

```
model = TSNE (perplexity= 30 , n_iter= 5000 , random_state= 0)
tsne_data = model.fit_transform(standarddata)
tsne_data.shape
```

Out[33]:

```
(42000, 2)
```

In [36]:

```
t_data = tsne_data
print (t_data)
```

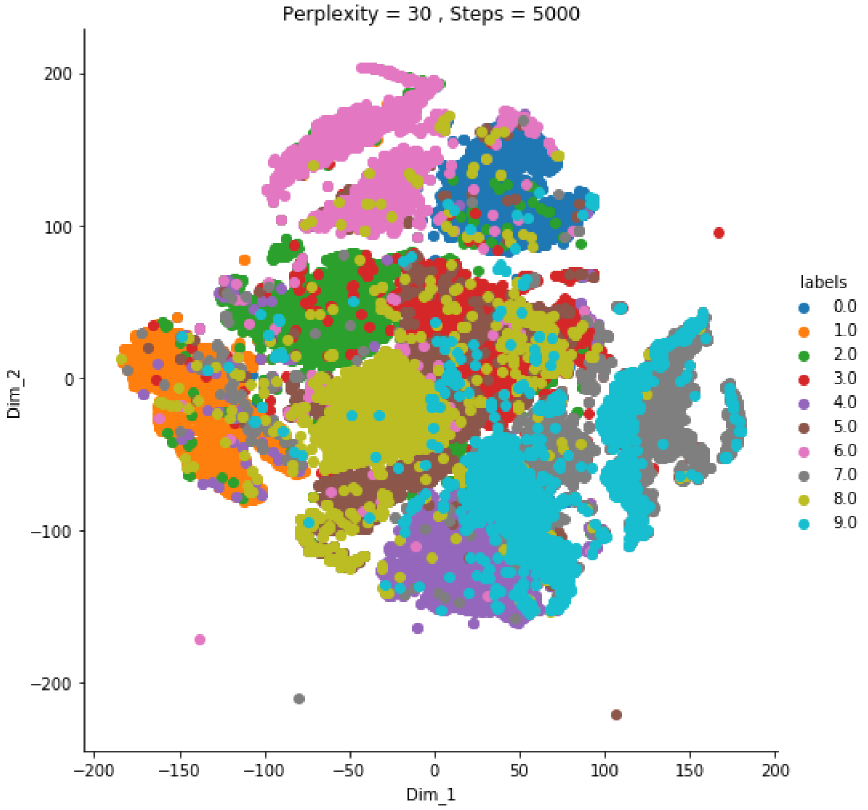
```
[[-106.60876    -67.68884  ]
 [  26.24263    133.31923  ]
 [-173.86064      3.1290288]
 ...
 [ 140.73538    -27.930756 ]
 [ -70.32488    152.4278   ]
 [  50.14733   -105.22112  ]]
```

In [37]:

```
import seaborn as sn
tsne_data = np.vstack ((tsne_data.T , label)).T
```

In [43]:

```
tsne_df = pd.DataFrame(data = tsne_data, columns = ('Dim_1' , 'Dim_2' , 'labels') )
sn.FacetGrid(tsne_df , hue = 'labels' , size = 7).map(plt.scatter , 'Dim_1' , 'Dim_2').add_legend()
plt.title('Perplexity = 30 , Steps = 5000')
plt.show()
```



t-SNE process takes a bit longer to execute. A model with 5000 iterations would approximately take about 15-20 mins of time depending upon your system. This process should be countinued for other values of preplexity and iterations to obtain best results.