小作业汇报

孙泽昌 2018012258

数据降维及其可视化

题目要求

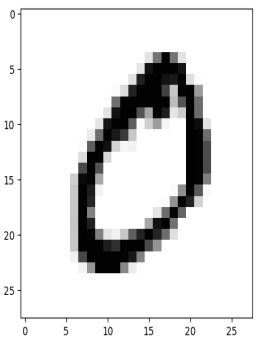
- 实现PCA算法
- 对结果使用散点图进行可视化分析
- 散点图可以进行简单交互

具体实现

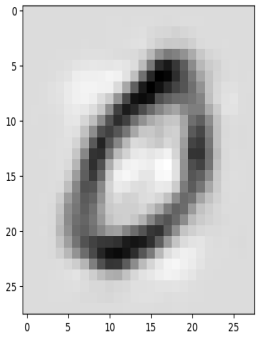
PCA算法实现

实现PCA算法首先需要将MNIST数据集中28×28维的2维数据展开成为784维的向量，之后将1000个张图片拼成一个输入矩阵X, 其形状为$(1000, 784)$，之后对$X^T X$进行本征值分解，这里调用`numpy.linalg.eig`函数实现。分解之后得到本征值Q和本征向量集V。综合可视化和数据重构率的要求对本征值向量和本征向量集进行截断，取截断后的本征值向量作为新的基V'，通过XV'^T将之前的高维数据投影到给定的新的d维空间中（具体实现中采用了$d = 2$和$d = 3$）得到降维之后的数据，此时PCA算法完成。可以通过观察取不同的截断维数时图片的重构情况来检查算法的正确性，如下图所示

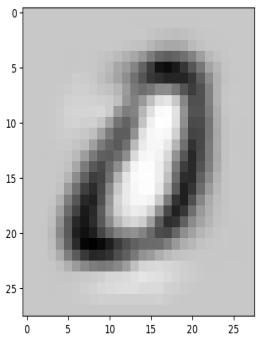
原始图片：



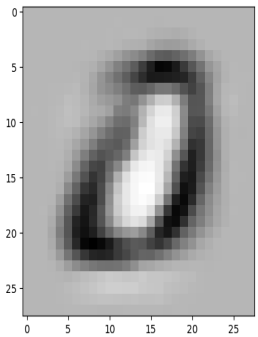
保留19个本征向量，重构率95.4%：



保留3个本征向量，重构率57.2%:



保留2个本征向量，重构率43.3%:



对结果的可视化分析

通过散点图可以比较清楚地展示经过降维后的数据分布情况，实现过程中采用了`pyecharts`来实现散点图以及简单的交互功能。由于绘制散点图本身不需要很大的绘图灵活性，而$d3.js$使用相对复杂，因此选用封装较好的`pyecharts`来实现。分别绘制了数据降维到2维后的散点图、数据降维到3维后，通过点的大小来表示第三个维度从而在二维展示的散点图以及三维散点图，绘图结果参见`PCA_visualization.html`。绘图的过程主要调用了`pyecharts.Scatter`，通过设置其中的参数来添加图例、交互操作等等，详细情况参见`PCA\plot.py`。

散点图的交互操作

绘制的散点图通过`pyecharts`包含了以下功能:

- 点击相应的点可以显示该点的详细信息;
- 图像的放大缩小操作;
- 三维散点图的旋转拉伸操作;

- 点击对应的图例可以隐去或显示对应类别的点。

结果讨论

从最后的结果图可以看出，通过`PCA`降维，数据在几乎保持原有特征的情况下能够在低维空间中进行可视化，但是PCA降维后得到的数据分界并不是十分明确，经常是很多不同类的数据混杂在一起，这也表明了PCA降维的效果并不尽人意。

我们使用`pyecharts`绘制了三种不同的散点图来进行数据可视化，可以发现进行适当的数据可视化之后，PCA降维后的效果得到了更加清楚的展示，体现了数据可视化的重要作用。

Density Map

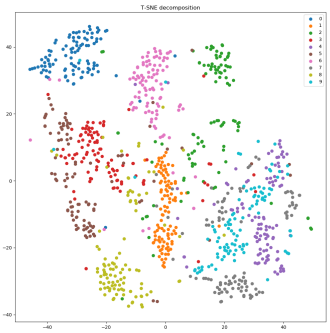
题目要求

- 将数据降维得到二维散点图；
- 使用Density map展示；
- 对散点图进行可视化展示；
- 对比不同核函数以及参数对density map的影响。

具体实现

数据降维

根据题目所述，可以调用降维算法API来实现，因此我们调用了`sklearn.manifold.TSNE`来实现，设置`n_components = 2`，可以得到降维到2维的散点图，使用`matplotlib`绘图展示如下：



可以看出t-SNE降维算法的表现优于PCA算法，降维后不同类的数据分开的程度明显更好。

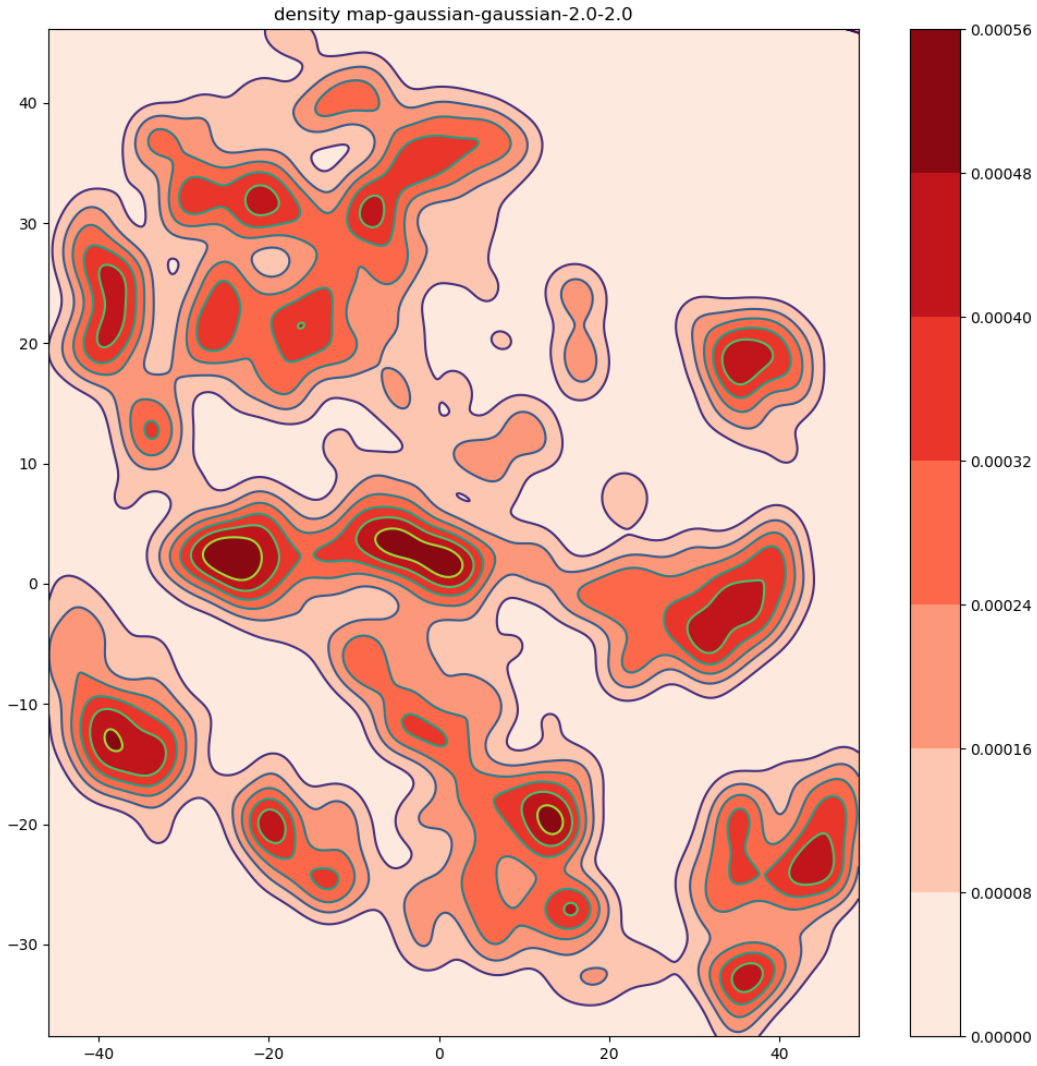
Density Map进行展示

参考https://blog.csdn.net/ilike_program/article/details/84978637，我构建了Density Map的算法，并将其进行可视化展示。具体实现如下： 首先选取核函数 （实现过程中采用了**Gaussian kernel**和**Epanechnikov kernel**），之后按照公式（这里我根据Density Map的含义对文献中的公式做了一些推广）：

$$p(x,y)=\frac{1}{N}\sum_{k=1}^{N}\frac{1}{h_1}K_1(\frac{x-x_k}{h_1})\cdot\frac{1}{h_2}K_2(\frac{y-y_k}{h_2})$$

按照Density Map的原理，二维情况下可以找一个二维的分布函数，这里为了简单方便起见，我们直接认为x, y方向分布独立，因此整体的二维kernel就为两个一维kernel的乘积。

计算概率密度分布，之后使用`matplotlib.pyplot.contour`进行可视化展示，得到的Density Map如下图所示。



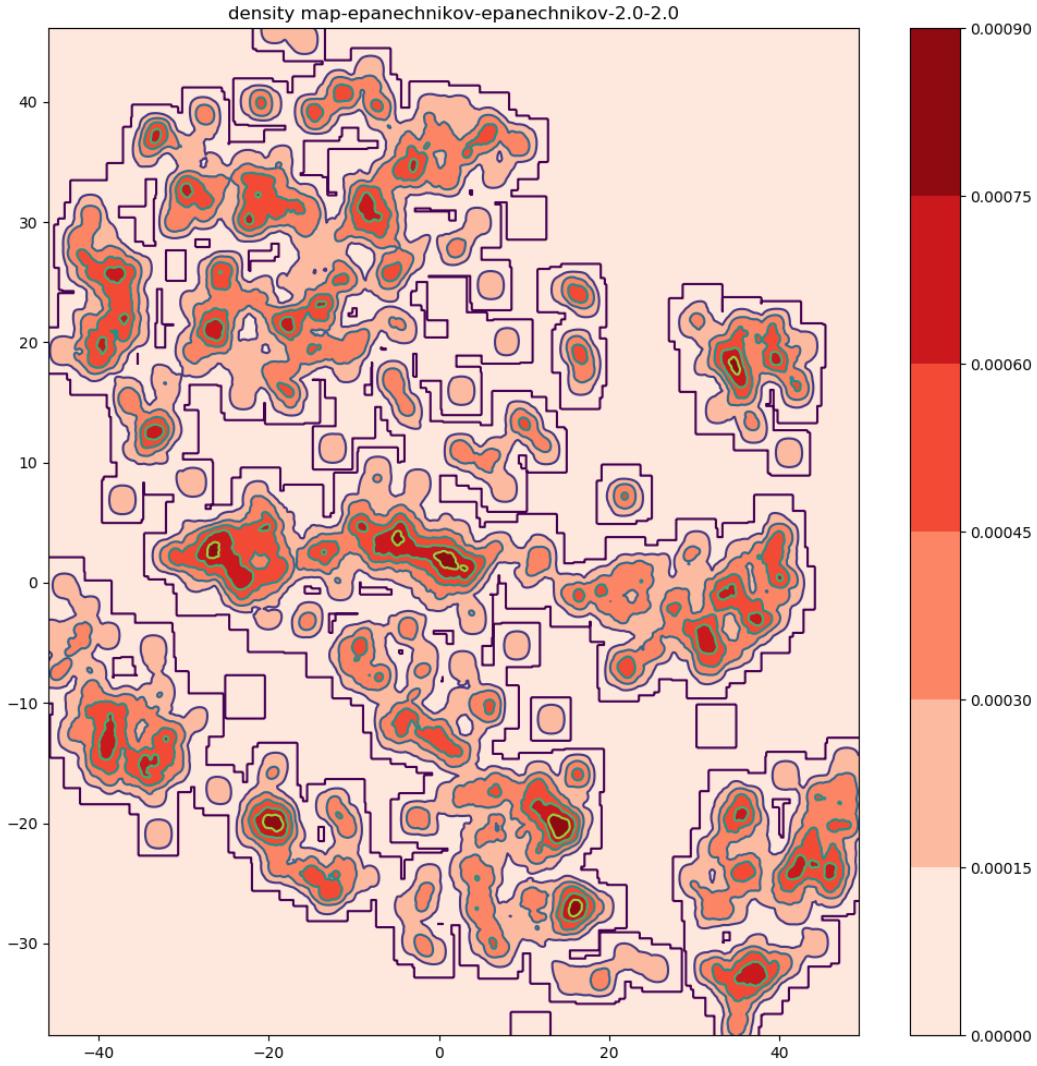
可以发现图中的聚类特征十分明显，也与上面的散点图相吻合。

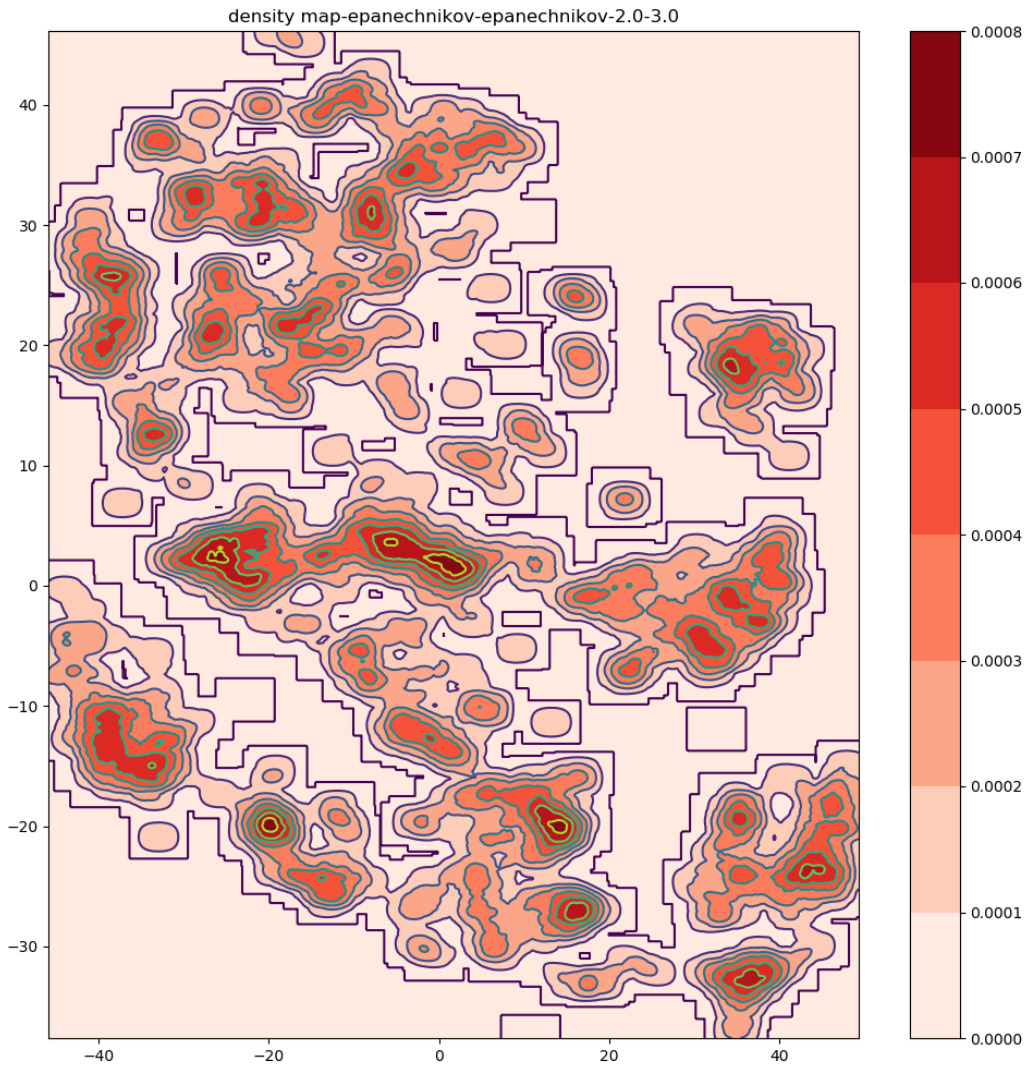
对散点图进行展示

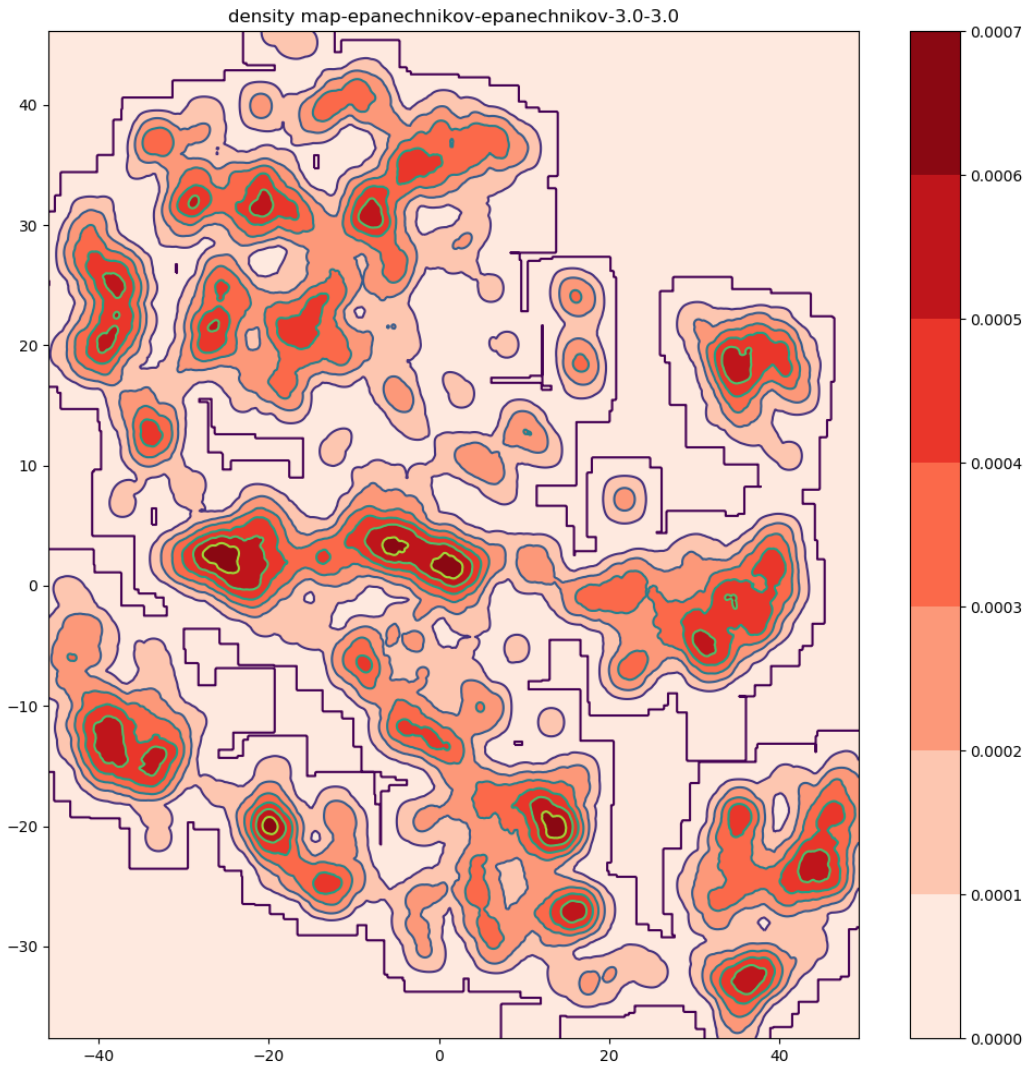
上面已经展示了数据降维后的散点图，这里就不再赘述。

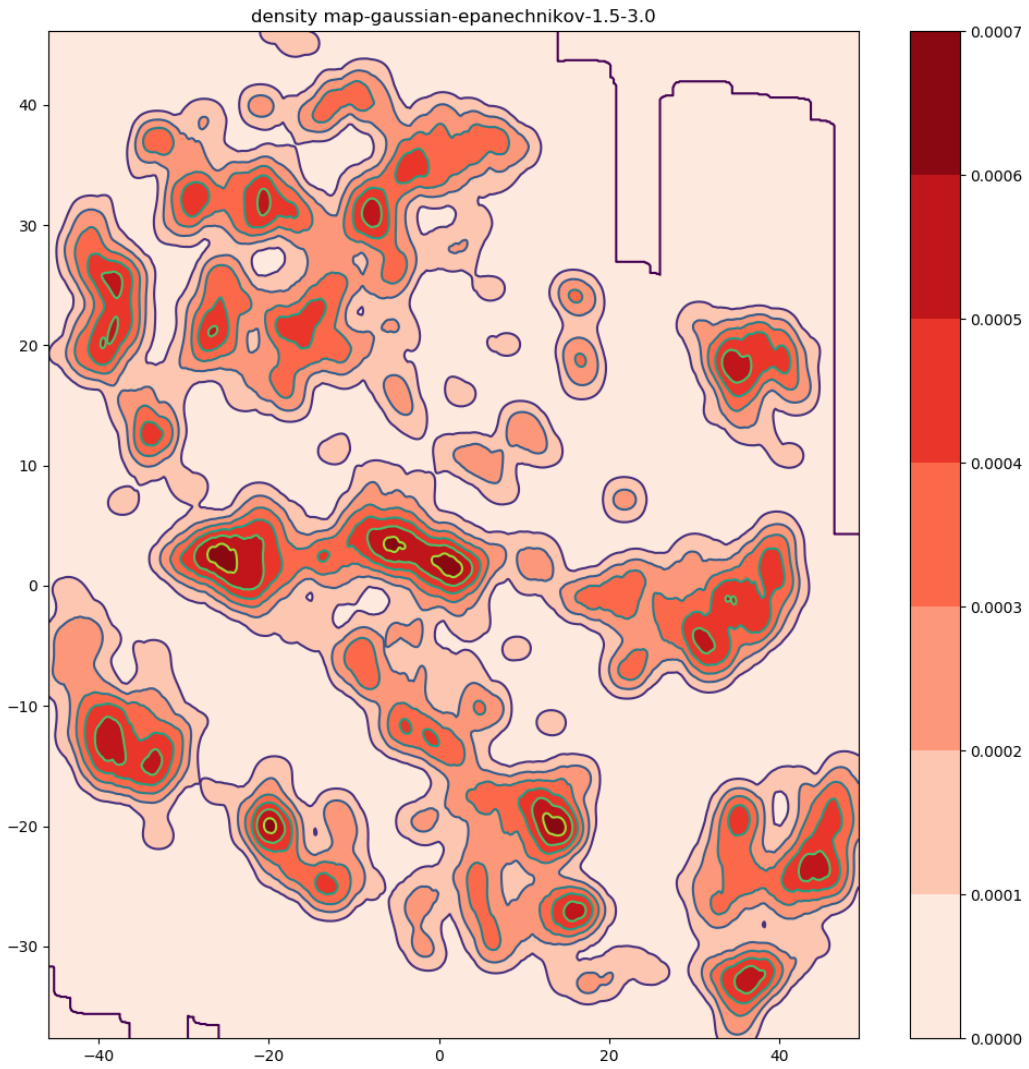
不同的核函数和带宽对Density Map的影响

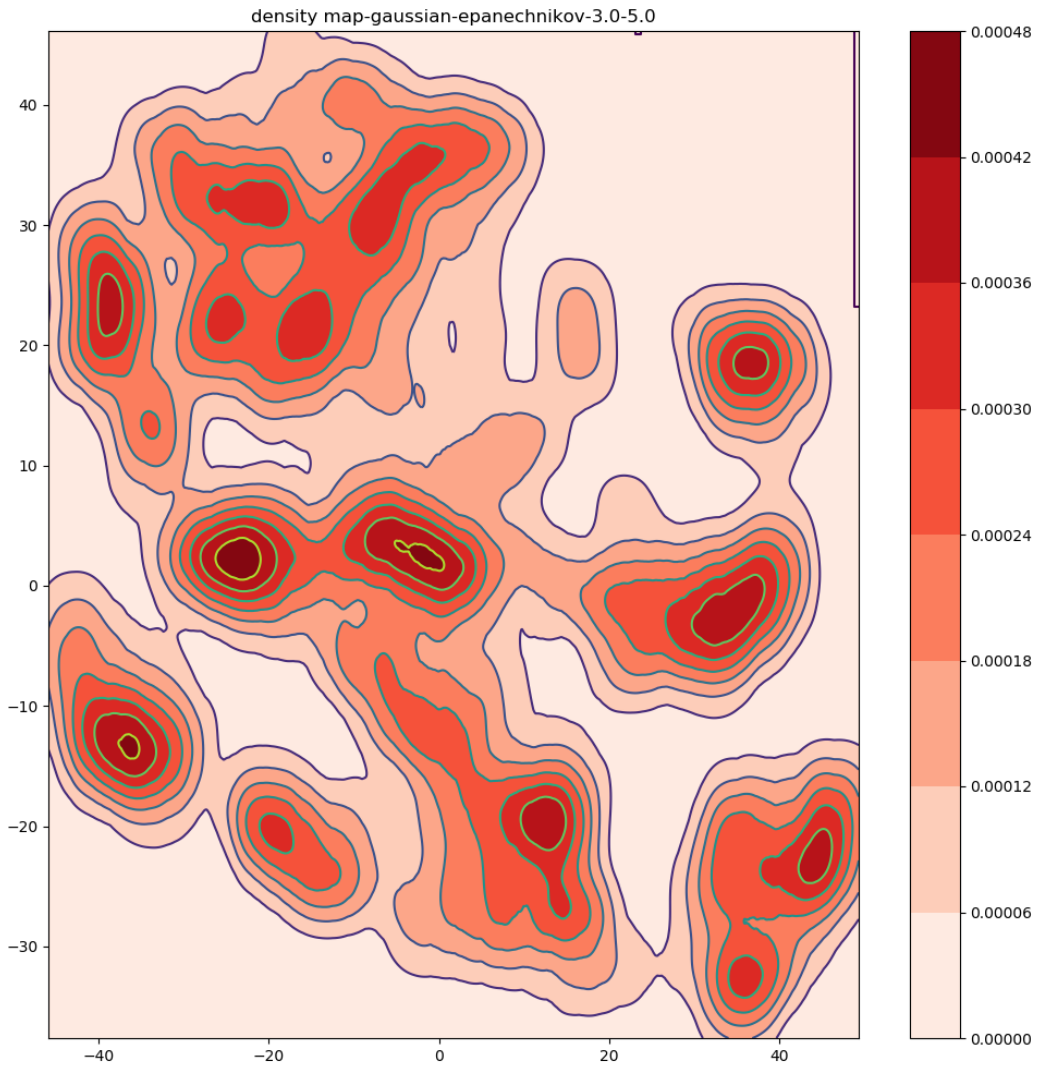
取不同的核函数和带宽，得到下面的散点图：

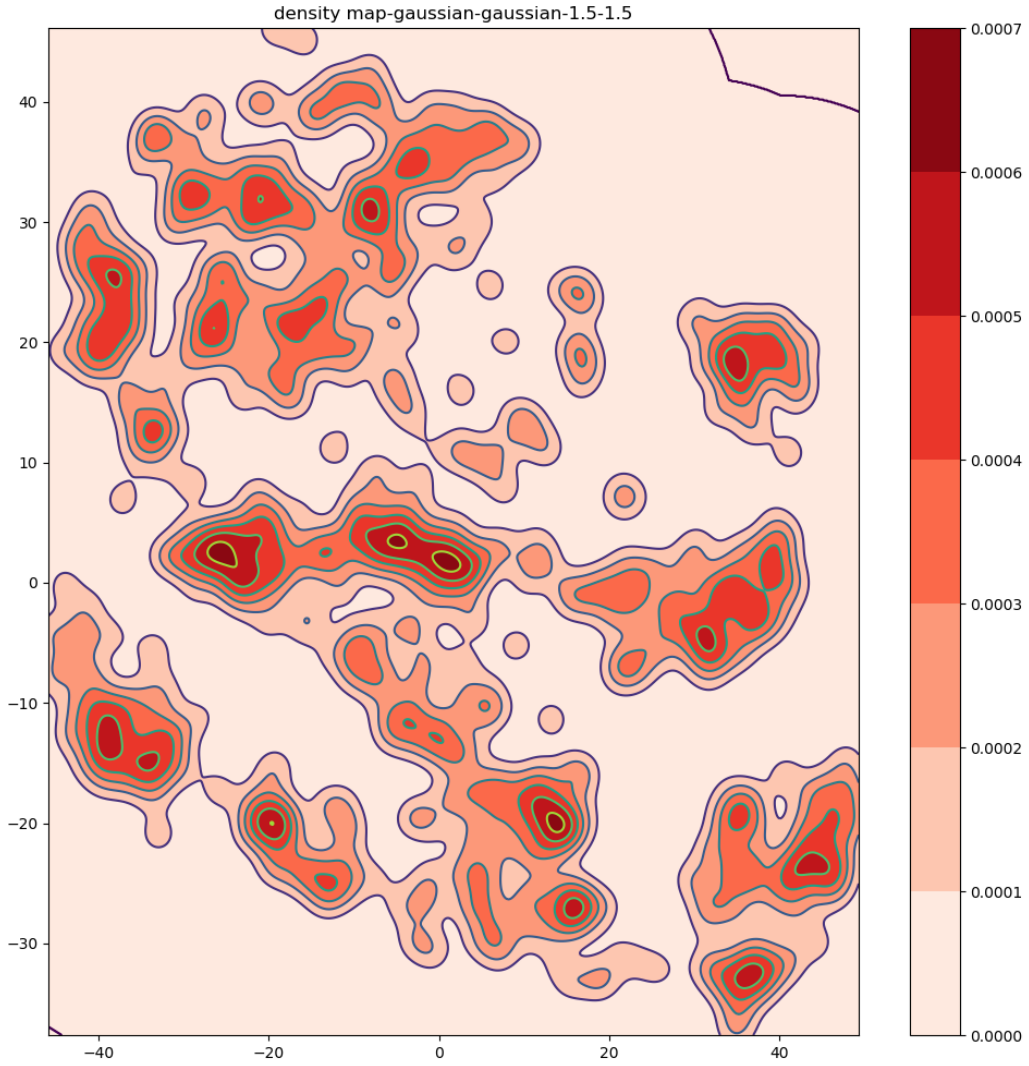

density map-epanechnikov-epanechnikov-2.0-2.0
40
30
20
10
0
−10
−20
−30
−40
−20
0
20
40
0.00090
0.00075
0.00060
0.00045
0.00030
0.00015
0.00000

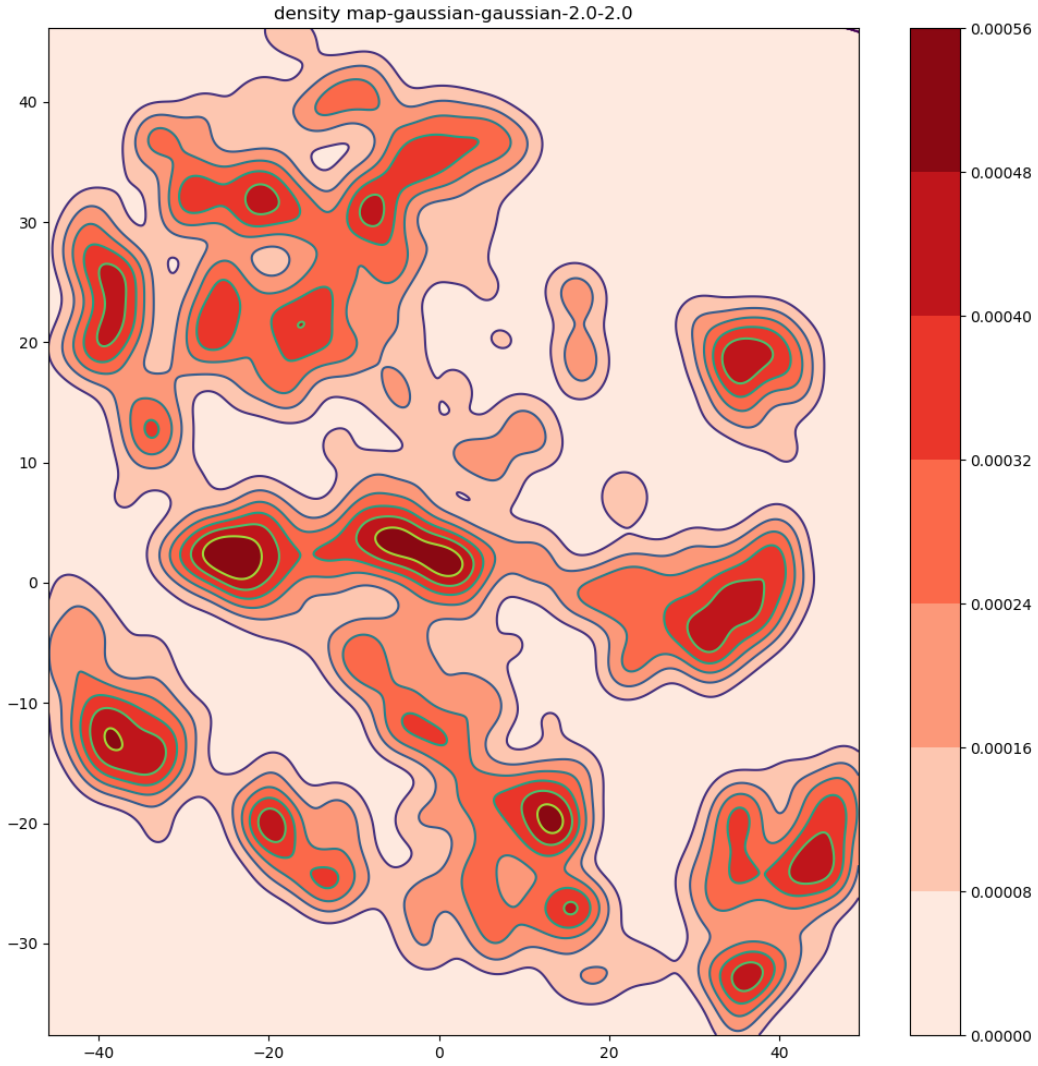

density map-epanechnikov-epanechnikov-2.0-3.0
40
30
20
10
0
−10
−20
−30
−40
−20
0
20
40
0.0008
0.0007
0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000

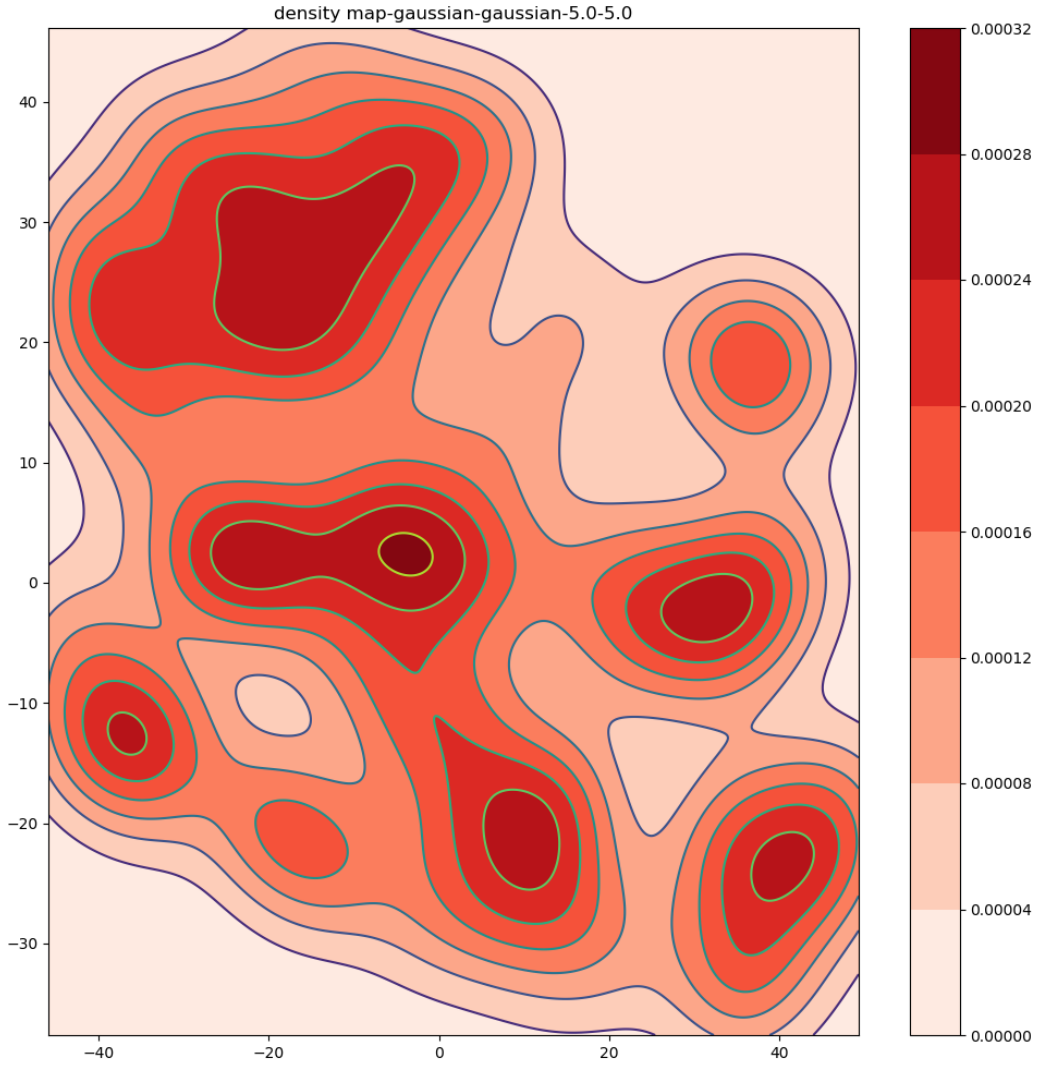

density map-epanechnikov-epanechnikov-3.0-3.0
40
30
20
10
0
−10
−20
−30
−40
−20
0
20
40
0.0007
0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000


density map-gaussian-epanechnikov-1.5-3.0
40
30
20
10
0
−10
−20
−30
−40
−20
0
20
40
0.0007
0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000


density map-gaussian-epanechnikov-3.0-5.0
40
30
20
10
0
−10
−20
−30
−40
−20
0
20
40
0.00048
0.00042
0.00036
0.00030
0.00024
0.00018
0.00012
0.00006
0.00000


density map-gaussian-gaussian-1.5-1.5
40
30
20
10
0
−10
−20
−30
−40
−20
0
20
40
0.0007
0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000


density map-gaussian-gaussian-2.0-2.0
40
30
20
10
0
−10
−20
−30
−40
−20
0
20
40
0.00056
0.00048
0.00040
0.00032
0.00024
0.00016
0.00008
0.00000



可以看出Epanechnikov kernel和Gaussian kernel在适当的参数下最后得到的Density Map是相似的，不同的点在于在点比较稀疏的区域，Gaussian kernel得到的等概率曲线仍然为曲线，而Epanechnikov kernel得到的等概率曲线为直线，同时也可以看出当h_1和h_2较大的时候，两种kernel画出的density map细节部分都会变得不明显，当h_1和h_2较小时，density map又失去了全局的整体特征，逐渐退化为散点图。

结果讨论

我们首先使用t-SNE算法将784维的数组降维到2维，之后使用kernel density estimation的方式估计所得到的降维后的点的密度分布并进行可视化。首先验证了t-SNE在数据降维方面的优异表现，之后探究了density map在数据可视化中的应用，以及不同的kernel和参数对density map的影响，总体而言可视化使得数据的特征被更好地展现，也使人对之后的分类问题等充满信心。