

系统工程方法在交通数据处理的应用

系统工程导论 课程设计

自 54 叶沁媛 2015011469

摘要 使用系统工程导论课程中学习的黑箱建模、主成分分析、聚类方法对交通流量数据进行处理和分析，主要应用为：利用历史数据预测未来的交通流量、利用主成分分析的结果描述一日内交通流量数据的特点、利用聚类方法和早高峰流量数据对路口进行分类。此外，通过自学掌握神经网络拟合和 SOM 聚类方法，与课上学习的传统方法进行了算法和结果上的比较。所给的数据具有三个维度（路口编号、时间、日期），这与往常我们所接触的数据有所不同，因此在选取样本的特征时综合考虑了生活经验和模型的特点，构思出多种可能有效的数据选取方法，以子任务的形式出现在各个任务中。通过对各个子任务的比较，可以验证猜想，并选择出更加合适的模型。

任务一：黑箱建模

注意到数据的特点（被预测的两天是周末），从生活经验的角度出发，我认为应当利用交通数据的周期性——使用本路口上周六到本周五的时段流量来预测本周六、日的同一时段流量。使用公式来描述：

$$\hat{y}_1 = \theta_{10} + \theta_{11}x_1 + \theta_{12}x_2 + \cdots + \theta_{17}x_7$$

$$\hat{y}_2 = \theta_{20} + \theta_{21}x_1 + \theta_{22}x_2 + \cdots + \theta_{27}x_7$$

其中 \hat{y}_1 与 \hat{y}_2 分别表示本周六、日，某一时段、某一路口的交通流量， x_1 至 x_7 分别表示这一时段、这一路口，上周六至本周五的交通流量。

构建模型的时候，我还思考了这样一个问题：对于每一个路段，应该各自建立黑箱模型预测，还是对 50 个路口建立一个整体的黑箱模型？它们的效果有多大的差别？计算复杂度又有多大的差别？对此，我设计了两个子任务。

子任务 1	对 50 个路口建立统一的黑箱模型
子任务 2	对每个路口建立独立的黑箱模型

子任务 1

以时段长度为 5 分钟为例，回归结果如下表所示：

	θ_0	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7
\hat{y}_1	20.59	0.32	0.36	0.04	0.05	0.04	0.06	0.07
\hat{y}_2	16.99	0.36	0.41	0.04	0.02	0.03	0.01	0.04

可以看到，和事先的判断一致，上周六、上周日的流量所乘系数较大，可以认为对预测值的影响较大。

在网上查找资料，得到平均绝对误差百分比（MAPE）和平均相对误差的（MRE）的表达式：

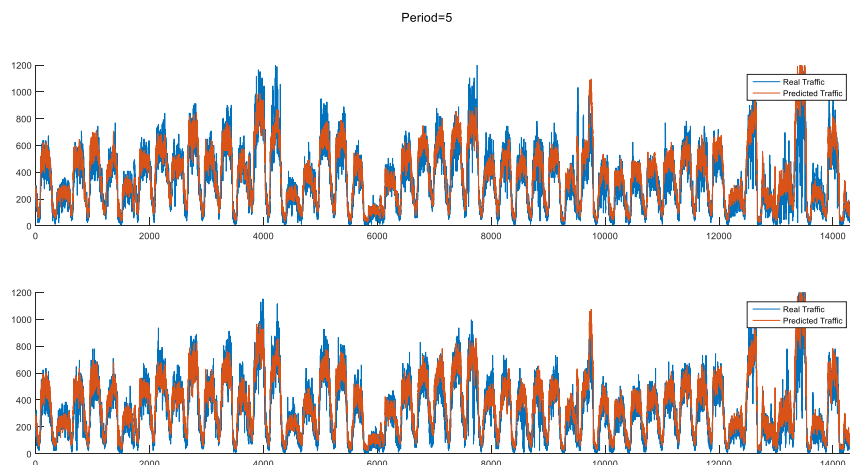
$$\text{MAPE} = \text{sum} \left(\frac{|\hat{y} - y^*|}{y^*} \right) / n$$

$$\text{MRE} = \text{sum} \left(\frac{\hat{y} - y^*}{y^*} \right) / n$$

对不同长度的时段分别进行测试，将结果列于下表。

时间长度		5 分钟	10 分钟	15 分钟
周六	平均绝对误差百分比	0.3744	0.2755	0.2424
	平均相对误差	0.2758	0.2063	0.1867
周日	平均绝对误差百分比	0.4339	0.3289	0.2921
	平均相对误差	0.3441	0.2681	0.2428
回归所用时长		0.133521	0.126733	0.105884

将预测结果和真实值绘制在一张图上进行比较。每一个图分为两个子图，上半部分代表所预测的周六交通流量，下半部分代表所预测的周日交通流量。由于周末没有早高峰，所以一天的交通流量大致是一个峰的形状（凌晨和深夜车流较小，白天车流较大），将 50 个路口在一天中的交通流量（即 50 个峰）连在一起进行显示。下图是时间长度为 5 分钟时的预测结果与真实值对比：



预测结果大致符合实际规律，但从 MAPE 和 MRE 的结果来看，仍有较大的误差。试分析误差存在的原因：

- ① 交通流量数据本来就存在很多不确定性，本身包含方差很大的噪声；
- ② 使用简单的线性回归难以捕捉到一些复杂的规律；

此外，MRE 的结果表明预测数据平均而言大于实际数据，这可能是由于一些使用模型无法表达的因素所决定的。例如，训练集中的周六周日天晴，测试集中的周六周日下雨，所以预测数据普遍大于真实数据（仅是猜测，2006 年的天气数据已无法找到）。

同时，还从误差的结果中发现，随着预测时段的增长（从 5 分钟变为 10 分钟和 15 分钟），MAPE 和 MRE 逐渐下降。这是因为，将 15 分钟看作了 3 个 5 分钟的叠加，时段变长实际代表着某种平均操作，故误差下降。

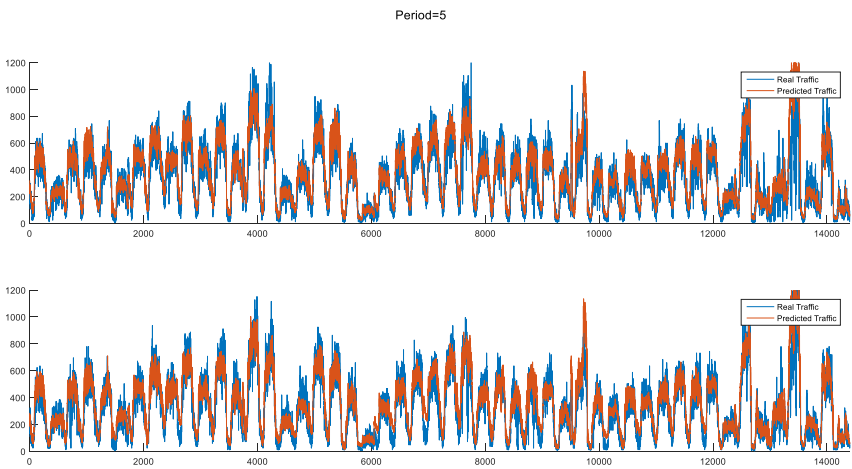
子任务 2

考虑到每个路口的特性是不同的，对于 50 个路口采用统一的模型可能无法提取出某些路口独特的特性，从而导致误差。在子任务 2 中，我们对于每一个路口分别做回归。

具体而言，对于某个路口某个时刻，过去 7 天的流量视为一个样本，若时长为 5 分钟，一次回归有 288 个样本；若时长为 15 分钟，则有 96 个样本。回归结果如下：

时间长度		5 分钟	10 分钟	15 分钟
周六	平均绝对误差百分比	0.3712	0.2722	0.2387
	平均相对误差	0.2666	0.1955	0.1744
周日	平均绝对误差百分比	0.4506	0.3462	0.3040
	平均相对误差	0.3420	0.2648	0.2328
回归所用时长		0.164319	0.186858	0.207038

下图是在子任务 2 中，时间长度为 5 分钟时的预测结果与真实值对比：



除了周日的 5 分钟平均误差百分比之外，都是子任务 2 所预测的结果优于子任务 1。可以认为对于每个路口单独进行回归可以得到更好的预测效果。但这一优势并不明显。并且，由于进行了多次回归，子任务 2 程序运行时间有所增加。

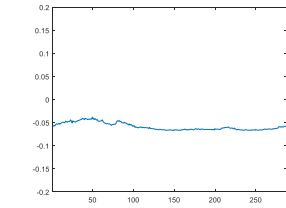
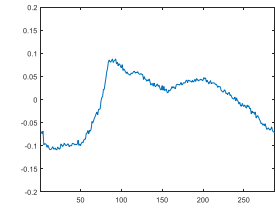
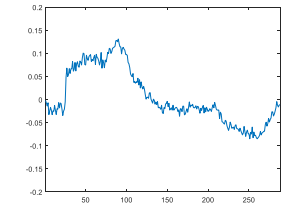
由于线性回归主要使用矩阵运算，在 Matlab 中矩阵运算经优化非常快速，所以回归时间均小于 0.25 秒，从使用者的角度而言没什么差异。但是当数据规模进一步增大时，后者的运行时长可能称为其明显的劣势。子任务 1 和子任务 2 实际是运算量和预测精度之间的一种 trade-off，实际使用时需要根据使用者的偏好来选取。

任务二：主成分分析

考虑到在一天内的交通流量会有一些相似性（如早高峰、晚高峰），将数据重组为 800*288 的格式，即每一个样本是某一个路口在某一天的 288 个 5 分钟间隔所测得的数据，并进行主成分分析。在保留 90%、95%、99% 的数据 Variance 的条件下，分别运行程序，得到如下结果：

保留 Variance	90%	95%	99%
主成分个数	59	119	221
压缩率	28.11%	56.44%	104.61%
RMSE	59.4328	44.1104	20.3818

取前三个主成分进行分析。将特征向量使用 plot() 命令绘制，帮助更加直观地体会其含义。

特征值	146050	20130	6497
特征向量			
含义解释	不同路段的日均车流不同。例如主干道的车流量较高，于是在这个特征向量上的系数为负；如果是比较偏僻的小道，日均车流较小，在这个特征向量上的系数为正。	在上午 7:00 和傍晚 17:00 左右形成了两个峰值，分别代表了早高峰和晚高峰；其中早高峰更加明显。	在上午形成了峰，在傍晚之后形成了谷；人们早上从住处前往市中心上班，夜间从市中心回到住处，这类车辆集中的路段会表现出这种规律。（数据记录的是特定方向的车流，而不是双向车流）

可以看到，前三大的特征值所对应的特征向量都能够根据生活经验得到很好的解释，这正是主成分分析的特点。第三大的特征值已经是第一大的特征值的约 1/20 了，这说明不同检测器在不同日期的平均值的差异很大，第一个特征很强；相比而言，早高峰、晚高峰的特征会稍弱一些。

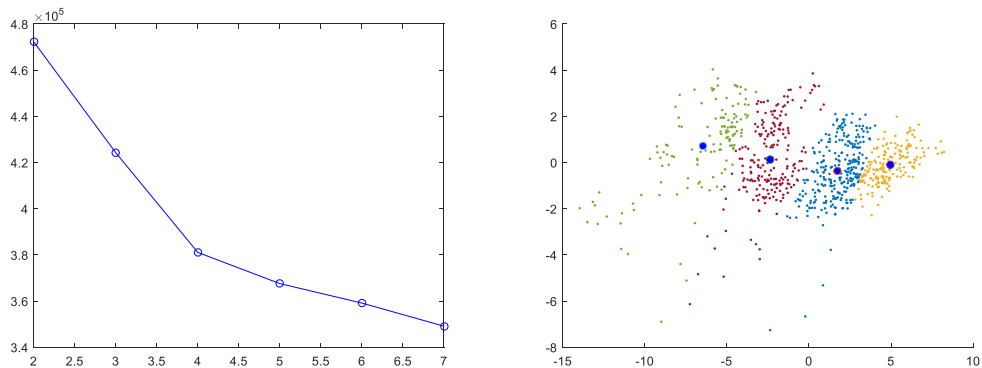
任务三：聚类分析

使用聚类分析，对早高峰时段（7 点至 9 点）各个路口的交通流量进行聚类分析。早高峰时段（7 点至 9 点）对应的是第 73-97 个数据点（每天每个路口有 288 个数据点，取出 24 个数据点）。在这个任务中，我设计了两个子任务：

子任务 1	以一个路口在某一天的 24 个数据点作为一个样本，共 $16 \times 50 = 800$ 个样本
子任务 2	以一个路口在 16 天的 16×24 个数据点作为一个样本，共 50 个样本

子任务 1

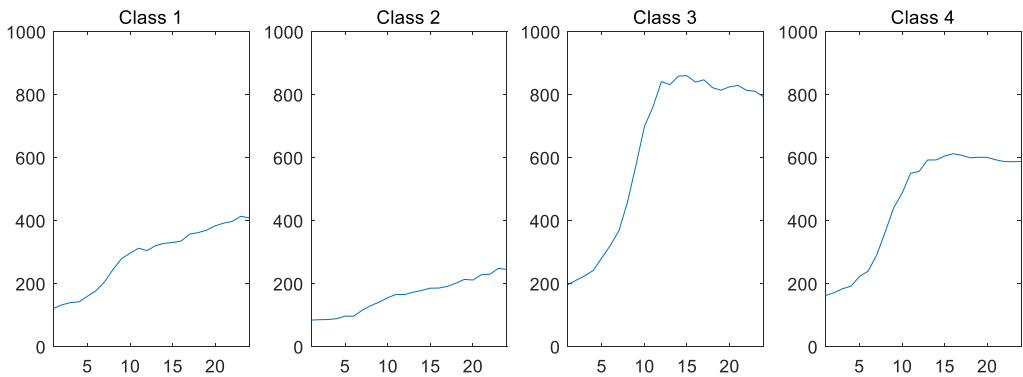
聚类数 n 从 2-7 枚举，绘制目标函数与聚类数 n 的关系图。在 $n > 4$ 以后，目标函数的下降不再明显。认为选择 $n = 4$ 最合适。



将聚类结果使用 PCA 降至二维以可视化。可以看到，数据被有效地分为四类。

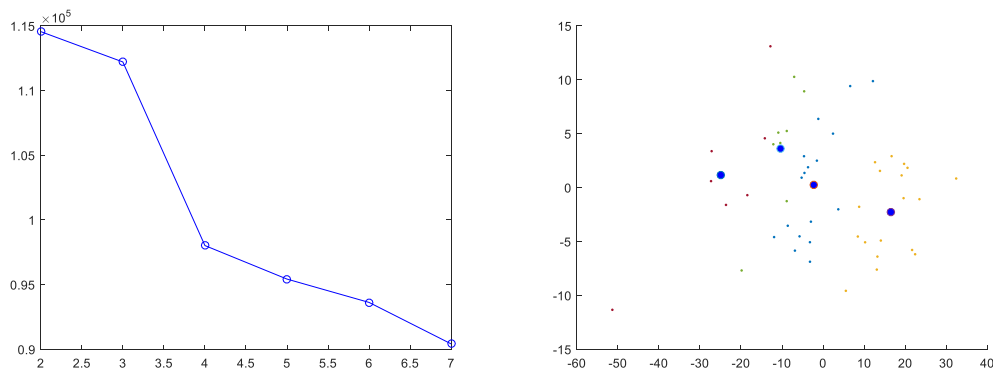
观察降维后的聚类结果，注意到这些类之间也没有特别明显的“留白”，这可能是降维可视化而导致的，在高维空间，这些类别之间会有更加明确的界限；这也可能是各类之间本身没有特别明确的分界所导致的。

绘制出四类的类中心所对应的早高峰交通流量，一定程度上代表了四类的特征。可以看到，这四类基本是按照早高峰的强烈程度来分的。其中第二类没有明显的早高峰；第四类的早高峰最为明显，交通流量在 7:30-8:00 之间显著上升。

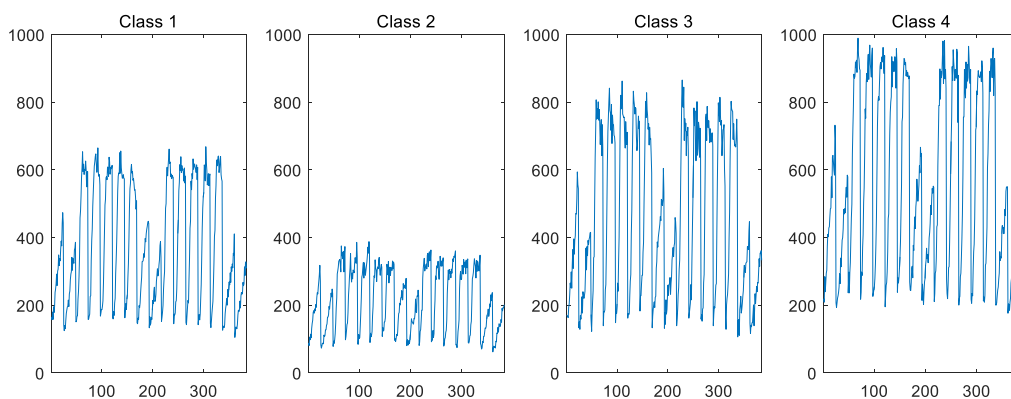


子任务 2

与子任务 1 类似，首先选择最合适的聚类数。在子任务 2 中，最合适的聚类数 n 仍为 4。由于在子任务 2 中，样本数仅为 50，因此聚类结果显得比较稀疏。



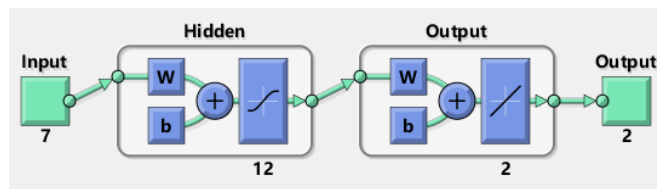
选取四类的中心点所对应的交通流量，可以看到数据呈现出了强烈的周期性，能看出数据的 16 天是 2 天周末+5 天工作日+2 天周末+5 天工作日+2 天周末的组合。第二类并没有明显的工作日早高峰，工作日的交通流量略大于周末；剩余三类的工作日交通流量均明显地大于周末；第四类代表了最繁忙的路段，在早高峰开始前的车流量就比其他三类的路段高，在早高峰最严重时的车流量更大达到了 900 以上。



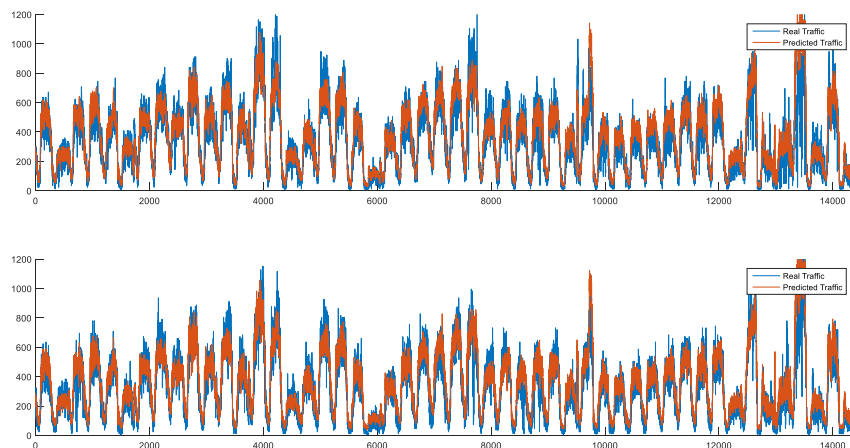
任务四：神经网络

从理论上可以证明，含有一个隐层的神经网络足以拟合任何函数¹。在任务一的基础上，我们使用含有一个隐层的神经网络对交通流量数据进行黑箱建模，所选取的隐层节点个数为 12。

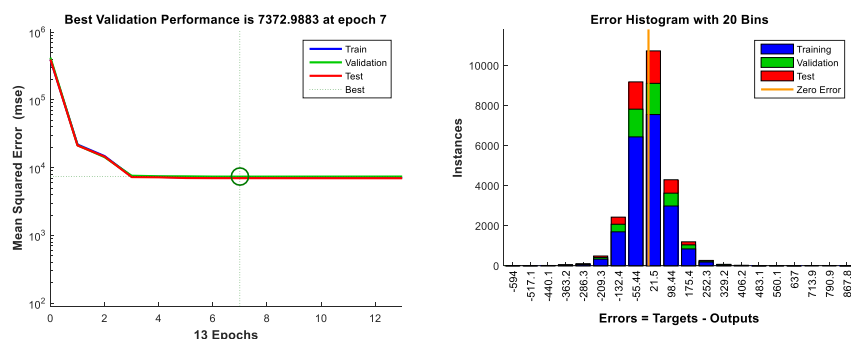
¹ A visual proof that neural nets can compute any function: <http://neuralnetworksanddeeplearning.com/chap4.html>



任务一分为 2 个子任务，其中第二个子任务需要进行 50 次回归。出于时间的考虑，这里仅进行子任务 1。时间长度为 5 分钟的预测结果如下：



同样以时间长度 5 分钟为例，神经网络训练了 13 个 Epochs 之后便收敛；训练集误差呈现一种近似正态分布的结构，大部分训练样本的误差均控制在 ± 100 的范围内。



对不同长度的时段（5 分钟、10 分钟、15 分钟）分别进行测试，将结果列于下表。绿色标注表示误差比任务一中的子任务 1 更小，红色标注表示更大。

时间长度		5 分钟	10 分钟	15 分钟
周六	平均绝对误差百分比	0.3498	0.2653	0.2273
	平均相对误差	0.2432	0.1947	0.1665
周日	平均绝对误差百分比	0.4283	0.3384	0.3136
	平均相对误差	0.3387	0.2778	0.2630
回归所用时长		3.654669	3.741786	3.485771

绿色的标注较多，看起来神经网络的误差更小，差别并不大；但是要注意到神经网络算法具有一定随机性，不能就此下结论。不过，我们能够肯定，神经网络训练时长远大于线性回归，时间上差 30 倍左右。

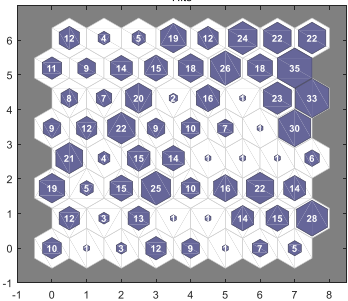
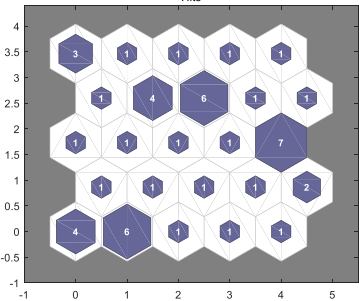
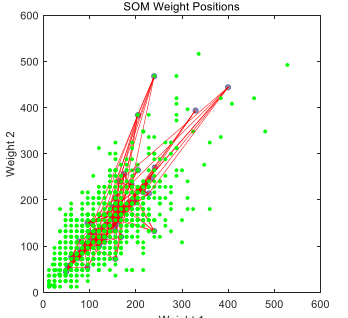
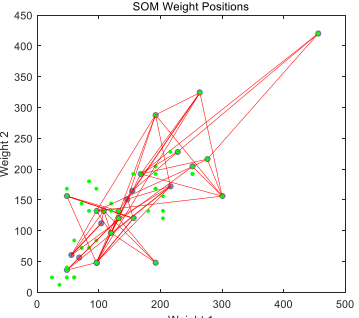
线性回归，以其简洁性和可解释性，在这个任务上颇具优势。由于模型设计的问题，预测值和输入样本之间本身的线性关系比较显著，且每一个样本的维数较小，神经网络此时可能无法发挥出其优势；在一些更加复杂的预测模型中（例如手写数字识别），神经网络能够发挥出其强大的功能。

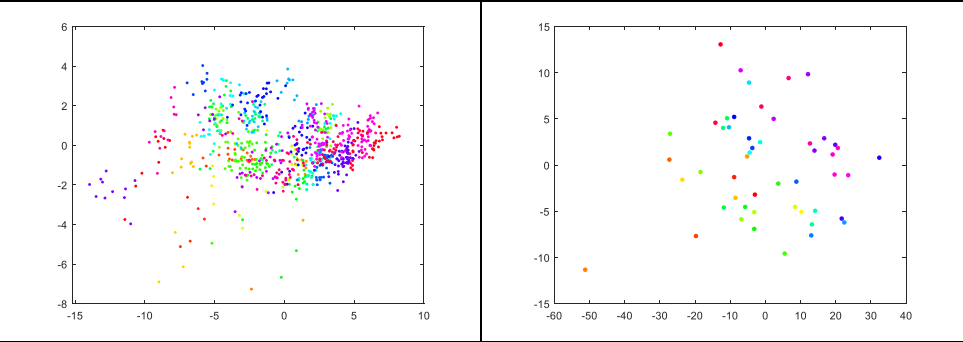
任务六：SOM 聚类方法

数据的预处理与任务三中相同，为了能够与任务三作比较，在任务六中也分别进行了子任务 1 和子任务 2。使用 Matlab 自带的 selforgmap()进行训练。由于子任务 1 和子任务 2 的样本数有巨大差别，经不断尝试并判断合理性，最终任务 1 选取的网格大小为 8×8 ，任务 2 选取的网格大小为 5×5 。具体代码如下：

```
net = selforgmap([8,8]);
net = train(net, data);
```

SOM 训练结果如下图所示：

	子任务 1	子任务 2
样本数量	800	50
输出层大小	8×8	5×5
象密度图 Sample Hits		
映射关系 Vector Position		

PCA 降维 可视化	
分析	在子任务 1 和子任务 2 中，样本都呈现出一些聚类的特性。子任务 1 中，在象密度图的(0,2),(3,2),(7,1),(7,5)位置均出现了较为明显的聚类，样本比较集中；子任务 2 中，在(1,0),(2,3),(4,2)位置出现了较为明显的聚类。
与任务三比较	<p>算法层面，K-means Clustering 是以最小化样本至类中心的距离和（一般为欧式距离）为目标，通过不断迭代（更新类中心、更新分类归属），最终收敛以达到聚类效果的。而 SOM 是更为高阶的一种聚类方法，其本质是一个两层的神经网络，第一层是输入层，第二层是二维输出层，每次迭代进行权值竞争学习，$m_i(t+1) = m_i(t) + \alpha(t)h_{ci}(t)d[x(t), m_i(t)]$，若干次迭代后结束。另一个区别是，K-means Clustering 需要预先指定聚类数n，而 SOM 则需要预先指定输出层大小，根据输出结果进一步合并小聚类得到最终的结果。</p> <p>结果层面，由于两个方法原理上的不同，所得的聚类结果也不相同。特别是 SOM 聚类方法，输出层大小的选取非常重要，在神经网络训练完成后还需要手动合并小聚类得到最后的结果。因此该算法具有一定的主观性。为了对两种聚类方法的结果进行比较，将聚类结果都在 PCA 降维后的二维平面上表示出。SOM 使用神经网络的模型，其中含有很多非线性的运算，在原空间的分类面可能是非线性的，所以在 PCA 降维后的二维平面上来看，部分类之间会有交叉。但 K-means Clustering 的分类取决于距离类中心点的距离，分类面是若干线性超平面的叠加，在 PCA 降维后的二维平面可以看出明确的类边界。由此得出两种方法结果不相同的结论。</p>

小结

通过本次课程设计，我将课程中学到的黑箱建模、主成分分析、聚类分析等方法应用到了交通流量预测和分析中。这些方法，或许只是在课堂中学习过推导，或许是在作业中进行了一些简单的应用；这次课程设计让我意识到他们在实际问题中的应用价值，并使我对这些方法的特点有了更加深刻的理解。

我还自学了神经网络、SOM 聚类方法，利用 Matlab 中自带的工具箱函数，对交通流量实现了预测和聚类，并和传统方法相比较。神经网络和 SOM 聚类其实在本质上有一些共通的地方，SOM 的模型本质是一个两层的神经网络；但他们却被用来实现了完全不同的功能（拟合和聚类）。这不禁让人感受到神经网络的魅力——只需要一点改进和创新，神经网络就可以被用来完成不同的任务。如今很多深度学习领域的科研工作者前赴后继，不断改进并发明新的神经网络结构，不仅是因为这些方法在

Performance 上的进步能够帮助解决更多现实问题、为人们的生活带来方便，更是因为神经网络本身的通用性和可塑性激励着科研工作者的想象力和创造力。

在观察程序运行结果的时候，我意识到，并不是越复杂的模型的预测结果越好。在任务一和任务四中，线性回归和神经网络所得到的误差指标相差无几。我将这归结于样本维数小的原因，以及线性模型本身适合求解这个问题的原因。在日后使用系统工程方法解决其他问题时，需要注意模型的选择，综合考虑精度和运算量等因素；可以尝试多种模型，并选择最合适的那一个。

附录：程序运行说明

任务名		源程序	备注
任务一	子任务 1	problem1.m	变量 period 表示预测时间长度， 可改为 5/10/15 中任意一个
	子任务 2	problem1_2.m	
任务二		problem2.m	
任务三	子任务 1	problem3.m	设置 flag=0
	子任务 2		设置 flag=1
任务四	子任务 1	problem4.m	同任务一
任务六	子任务 1	problem6.m	设置 flag=0
	子任务 2		设置 flag=1

源程序与本报告均已同步更新至 GitHub：<https://github.com/cherry979988/system-engineering-project>