数据挖掘项目总结报告

学校: 北京交通大学

专业: 软件工程

汇报人: 王梓铭

起止时间: 2018 年 2 月 20 日 至 2018 年 2 月 26 日

Contents

1 研究背景

1.1 数据挖掘背景

数据挖掘起始于 20 世纪下半叶，是在当时多个学科发展的基础上发展起来的。随着数据库技术的发展应用，数据的积累不断膨胀，导致简单的查询和统计已经无法满足企业的商业需求，急需一些革命性的技术去挖掘数据背后的信息。同时，这期间计算机领域的人工智能（Artificial Intelligence）也取得了巨大进展，进入了机器学习的阶段。因此，人们将两者结合起来，用数据库管理系统存储数据，用计算机分析数据，并且尝试挖掘数据背后的信息。这两者的结合促生了一门新的学科，即数据库中的知识发现（Knowledge Discovery in Databases，KDD）。1989 年 8 月召开的第 11 届国际人工智能联合会议的专题讨论会上首次出现了知识发现（KDD）这个术语，到目前为止，KDD 的重点已经从发现方法转向了实践应用。

而数据挖掘（Data Mining）则是知识发现（KDD）的核心部分，它指的是从数据集合中自动抽取隐藏在数据中的那些有用信息的非平凡过程，这些信息的表现形式为：规则、概念、规律及模式等。进入 21 世纪，数据挖掘已经成为一门比较成熟的交叉学科，并且数据挖掘技术也伴随着信息技术的发展日益成熟起来。总体来说，数据挖掘融合了数据库、人工智能、机器学习、统计学、高性能计算、模式识别、神经网络、数据可视化、信息检索和空间数据分析等多个领域的理论和技术，是 21 世纪初期对人类产生重大影响的十大新兴技术之一。

2 相关技术

本项目需要对获取的数据进行清洗，去噪声，聚类，预测等步骤. 在本次课程中老师讲解了一些相关的基础知识，以下是我对这些技术的总结.

2.1 主成分分析法

2.1.1 基本思想

主成分分析所要做的就是设法将原来众多具有一定相关性的变量，重新组合为一组新的相互无关的综合变量来代替原来变量。通常，数学上的处理方法就是将原来的变量做线性组合，作为新的综合变量，对于变量所包含的信息，这里用方差来测量，即：方差越大，则此变量包含的信息越多。因此在所有的线性组合中所选取的第一个变量应该是方差最大的，故称为第一主成分。如果第一主成分不足以代表原来所有变量的信息，再考虑选取方差第二大的变量即第二主成分. 但为了有效地反映原来信息，第一主成分已有的信息就不需要再出现在第二主成分中，因此要求各个主成分之间彼此独立. 依此类推可以构造出第三主成份、第四主成份等。

2.1.2 计算步骤

①对原始数据进行标准化处理。设有 n 个样本观察值，其中，x_{ij} 表示原样本指标 x_j 的第 i 个产业实际值，经过处理的数据的均值为 0，标准差为 1。转化公式为：

$$x_{ij}^{*} = \frac{x_{ij} - \overline{x_j}}{\sigma_j}$$

其中 $\overline{x_j}$ 为各指标原始数据的均值，σ_j 为各指标原始数据的标准差。但是当均值和标准差受离群点的影响很大时，用中值数 M 取代均值，其次用绝对标准差取

代标准差，$\sigma_j^* = \sum_{i=1}^{i=n} \left| x_{ij} - W \right|$，$W$ 是平均数或者中位数。

②计算 $\left[x_{ij}\right]_{n*p}$ 相关系数矩阵 R。相关系数矩阵 R 可以发现各测评指标间的相关状况，从而能够看出指标间的信息重叠程度。

$$R = (r_{jk})_{p*p} \quad (j=1,2,\ldots,p;\ k=1,2,\ldots,p)$$

其中相关系数为：

$$r_{jk} = \frac{1}{n-1} \sum_{i=1}^{n} x_{ij}^* x_{ik}^* \quad (i=1,2,\ldots,n;\ j=1,2,\ldots,p;\ k=1,2,\ldots,p)$$

③计算相关矩阵 R 的特征值、特征向量。

将得到的相关矩阵 R 的主对角线元素改为$(1-\lambda)$，构成矩阵行列式，令该行列式的值为 0，可以解出 λ_1，λ_2，$\ldots$，λ_p 的 P 个特征值。将特征值代入方程 $AX = \lambda X$，求解即可得到与特征值相对应的特征向量。由上述计算可知，λ_i 为第 i 个主成分 F_i 的方差，它的大小体现了各个主成分在描述被评价对象上所起作用的大小，从而确定主成分。

④计算累计方差贡献率，确定主成分个数。

根据各个指标的相关系数矩阵，得出主成分对总方差的累计方差贡献率。在确定主成分个数时，可以根据特征值的大小确定，一般取特征值 λ 大于 1 的主成分；或者是用累积方差贡献率 Q 确定，一般满足$80\% \leqslant Q \leqslant 95\%$，主成分的个数 m 值往往不超过 3。

$$Q = \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{p} \lambda_i} \quad (i = 1,2,\ldots,\ p)$$

⑤根据因子载荷矩阵，确定主成分得分系数矩阵。

根据已确定的主成分的因子载荷矩阵，计算已选出的 m 个主成分的单位特征向

量u_1，u_2,...，u_m，结合专业知识对所选择的主成分给予恰当的解释，并计算各指标变量对各个主成分的负荷量，设第i个变量（因子）对第j个主成分的负荷量为a_{ij}，则$a_{ij}=u_{ij}\sqrt{\lambda_i}$（$i$，$j$=1,2，...，p），从而求得主成分得分系数矩阵。

⑥计算综合评价权重值。

求出 m 个主成分的线性加权值，即

$$F_g=l_{g1}Z_1+l_{g2}Z_2+...+l_{gp}Z_p$$

$$l_g=\frac{\lambda_g}{\sum_{g-1}^{p}\lambda_g} \quad (g=1,2,...,\ m)$$

最终权重值为：

$$F=\sum_{g-1}^{m}\left(\frac{\lambda g}{\sum_{g-1}^{p}\lambda g}\right)F_g \quad (g=1,2,...,\ m)$$

2.2 非负矩阵分解

2.2.1 基本思想

非负矩阵分解由 Lee 和 Seung 于 1999 年在自然杂志上提出，它使分解后的所有分量均为非负值(要求纯加性的描述)，并且同时实现非线性的维数约减。NMF 的心理学和生理学构造依据是对整体的感知由对组成整体的部分的感知构成的(纯加性的)，这也符合直观的理解：整体是由部分组成的，因此它在某种意义上抓住了智能数据描述的本质．此外，这种非负性的限制导致了相应描述在一定程度上的稀疏性，稀疏性的表述已被证明是介于完全分布式的描述和单一活跃分量的描述之间的一种有效数据描述形式。

2.2.2 推导公式

简单来讲，非负矩阵分解是在矩阵分解的基础上对分解完毕的矩阵加上非负的限制条件。即对于用户-商品矩阵 Vm×n，找到两个矩阵 Wm×k 和 Hk×n，使得：

$$Vm\times n \approx Wm\times k \times Hk\times n = V\hat{}m\times n$$

同时要求：

$$Wm\times k \geqslant 0$$

$$Hk\times n \geqslant 0$$

2.2.3 损失函数与迭代函数

为了能够定量的比较矩阵 Vm×n 和矩阵 V^m×n 的近似程度。在参考文献 1 中作者提出了两种损失函数的定义方式：

平方距离

$$//A-B//\hat{}2=\sum i,j(Ai,j-Bi,j)2$$

当定义好损失函数后，须要求解的问题就变成了例如以下的形式，即求解例如以下的最小化问题：

$$minimize//V-WH//\hat{}2$$

$$s.t.W\geqslant 0,H\geqslant 0$$

在原文文献中，作者提出了乘法更新规则(multiplicative update rules)，详细的操作例如以下：

对于平方距离的迭代函数：

$$W_{ik} = W_{ik} \cdot \frac{(VH^T)_{ik}}{(WHH^T)_{ik}}$$

$$H_{kj} = H_{kj} \cdot \frac{(W^TV)_{kj}}{(W^TWH)_{kj}}$$

2.3 KNN – K 近邻算法

2.3.1 算法介绍

kNN 算法的核心思想是如果一个样本在特征空间中的 k 个最相邻的样本中的大多数属于某一个类别, 则该样本也属于这个类别, 并具有这个类别上样本的特性。该方法在确定分类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。 kNN 方法在类别决策时，只与极少量的相邻样本有关。由于 kNN 方法主要靠周围有限的邻近的样本，而不是靠判别类域的方法来确定所属类别的, 因此对于类域的交叉或重叠较多的待分样本集来说, kNN 方法较其他方法更为适合。

2.3.2 算法流程

准备数据，对数据进行预处理

选用合适的数据结构存储训练数据和测试元组

设定参数，如 k

维护一个大小为 k 的的按距离由大到小的优先级队列，用于存储最近邻训练元组。其中距离可定义为:

欧式距离： $d(x,y)=\sqrt{\sum_{k=1}^{n}(x_k-y_k)^2}$ ，　　曼哈顿距离： $d(x,y)=\sqrt{\sum_{k=1}^{n}|x_k-y_k|}$

然后随机从训练元组中选取 k 个元组作为初始的最近邻元组, 分别计算测试元组到这 k 个元组的距离，将训练元组标号和距离存入优先级队列.

遍历训练元组集，计算当前训练元组与测试元组的距离，将所得距离 L 与优先级队列中的最大距离 Lmax.

进行比较。若 L>=Lmax，则舍弃该元组，遍历下一个元组。若 L < Lmax，删除优先级队列中最大距离的元组，将当前训练元组存入优先级队列。

遍历完毕，计算优先级队列中 k 个元组的多数类，并将其作为测试元组的类别。

测试元组集测试完毕后计算误差率，继续设定不同的 k 值重新进行训练，最后取误差率最小的 k 值。

2.4 朴素贝叶斯算法

2.4.1 算法介绍与推导

对于给出的待分类项，求解在此项出现的条件下各个类别出现的概率，哪个最大，就认为此待分类项属于哪个类别。在没有其它可用信息下，我们会选择条件概率最大的类别，这就是朴素贝叶斯的思想基础。

朴素贝叶斯分类的正式定义如下：

设$x = \{a_1, a_2, ..., a_m\}$为一个待分类项，而每个 a 为 x 的一个特征属性。

有类别集合$C = \{y_1, y_2, ..., y_n\}$。

计算$P(y_1|x), P(y_2|x), ..., P(y_n|x)$。

如果$P(y_k|x) = max\{P(y_1|x), P(y_2|x), ..., P(y_n|x)\}$，则$x \in y_k$。

那么现在的关键就是如何计算第 3 步中的各个条件概率。我们可以这么做：

找到一个已知分类的待分类项集合，这个集合叫做训练样本集。

统计得到在各类别下各个特征属性的条件概率估计。即

$P(a_1|y_1), P(a_2|y_1), ..., P(a_m|y_1); P(a_1|y_2), P(a_2|y_2), ..., P(a_m|y_2); ...; P(a_1|y_n), P(a_2|y_n), ..., P($

如果各个特征属性是条件独立的，则根据贝叶斯定理有如下推导：

$$P(y_i|x) = \frac{P(x|y_i)P(y_i)}{P(x)}$$

因为分母对于所有类别为常数，因为我们只要将分子最大化皆可。又因为各特征

属性是条件独立的，所以有：

$$P(x|y_i)P(y_i) = P(a_1|y_i)P(a_2|y_i)...P(a_m|y_i)P(y_i) = P(y_i)\prod_{j=1}^{m} P(a_j|y_i)$$

2.4.2 算法流程

第一阶段——准备工作阶段，这个阶段的任务是为朴素贝叶斯分类做必要的准备，主要工作是根据具体情况确定特征属性，并对每个特征属性进行适当划分，然后由人工对一部分待分类项进行分类，形成训练样本集合。这一阶段的输入是所有待分类数据，输出是特征属性和训练样本。这一阶段是整个朴素贝叶斯分类中唯一需要人工完成的阶段，其质量对整个过程将有重要影响，分类器的质量很大程度上由特征属性、特征属性划分及训练样本质量决定。

第二阶段——分类器训练阶段，这个阶段的任务就是生成分类器，主要工作是计算每个类别在训练样本中的出现频率及每个特征属性划分对每个类别的条件概率估计，并将结果记录。其输入是特征属性和训练样本，输出是分类器。这一阶段是机械性阶段，根据前面讨论的公式可以由程序自动计算完成。

第三阶段——应用阶段。这个阶段的任务是使用分类器对待分类项进行分类，其输入是分类器和待分类项，输出是待分类项与类别的映射关系。这一阶段也是机械性阶段，由程序完成。

2.5 KMeans — K 均值算法

K-means 算法是最为经典的基于划分的聚类方法，是十大经典数据挖掘算法之一。K-means 算法的基本思想是：以空间中 k 个点为中心进行聚类，对最靠近他们的对象归类。通过迭代的方法，逐次更新各聚类中心的值，直至得到最好的聚类结果。

假设要把样本集分为 c 个类别，算法描述如下：

（1）适当选择 c 个类的初始中心；

（2）在第 k 次迭代中，对任意一个样本，求其到 c 个中心的距离，将该样本归到距离最短的中心所在的类；

（3）利用均值等方法更新该类的中心值；

（4）对于所有的 c 个聚类中心，如果利用（2）（3）的迭代法更新后，值保持不变，则迭代结束，否则继续迭代。

该算法的最大优势在于简洁和快速。算法的关键在于初始中心的选择和距离公式。

2.6 DBSCAN 算法

2.6.1 属性定义

E邻域：给定对象半径为E内的区域称为该对象的E邻域；

核心对象：如果给定对象E领域内的样本点数大于等于 MinPts，则称该对象为核心对象；

直接密度可达：对于样本集合 D，如果样本点 q 在 p 的 E 领域内，并且 p 为核心对象，那么对象 q 从对象 p 直接密度可达。

密度可达：对于样本集合 D，给定一串样本点 p1,p2…pn，p= p1,q= pn,假如对象 pi 从 pi-1 直接密度可达，那么对象 q 从对象 p 密度可达。

密度相连：存在样本集合 D 中的一点 o，如果对象 o 到对象 p 和对象 q 都是密度可达的，那么 p 和 q 密度相联。

可以发现，密度可达是直接密度可达的传递闭包，并且这种关系是非对称的。密度相连是对称关系。DBSCAN 目的是找到密度相连对象的最大集合。

Eg: 假设半径 E=3，MinPts=3，点 p 的 E 领域中有点{m,p,p1,p2,o}，点 m 的 E 领域中有点{m,q,p,m1,m2},点 q 的 E 领域中有点{q,m},点 o 的 E 领域中有点{o,p,s},点 s 的 E 领域中有点{o,s,s1}.

那么核心对象有 p,m,o,s(q 不是核心对象，因为它对应的 E 领域中点数量等于 2，小于 MinPts=3)；

点 m 从点 p 直接密度可达，因为 m 在 p 的 E 领域内，并且 p 为核心对象；

点 q 从点 p 密度可达，因为点 q 从点 m 直接密度可达，并且点 m 从点 p 直接密度可达；

点 q 到点 s 密度相连，因为点 q 从点 p 密度可达，并且 s 从点 p 密度可达。

2.6.2 算法流程

输入: 包含 n 个对象的数据库，半径 e，最少数目 MinPts;

输出:所有生成的簇，达到密度要求。

(1)Repeat

(2)从数据库中抽出一个未处理的点；

(3)IF 抽出的点是核心点 THEN 找出所有从该点密度可达的对象，形成一个簇；

(4)ELSE 抽出的点是边缘点(非核心对象)，跳出本次循环，寻找下一个点；

(5)UNTIL 所有的点都被处理。

DBSCAN 对用户定义的参数很敏感，细微的不同都可能导致差别很大的结果，而参数的选择无规律可循，只能靠经验确定。

3 项目方案与实现

3.1 项目问题背景

城市交通拥堵治理问题是当前我国乃至世界范围内的一个热门话题和难题。2000年,诺贝尔奖获得者加里・贝克尔做过一个测算,全球每年因拥堵造成的损失占GDP 的 215%。美国 2017 年拥堵成本 3050 亿美元，人均 1400 美元，洛杉矶全球最堵，人均损失 2800 美元. 伦敦每年因交通拥堵损失 55 亿英镑. 2017 全球最堵的是泰国，人均堵车 56 小时. 基于以上种种现象，有人认为,交通拥堵是一种现代城市病,极其复杂并难以解决。但实际上，拥堵总是存在的,只是有的拥堵可以接受,而有的拥堵则不可接受。本项目旨在通过对拥堵进行定义分类，预测并确定异常拥堵. 通过本项目的方法，城市交通管理者可以更好的对异常流量道路进行及时处理.

3.2 问题定义

3.2.1 交通异常定义

常规性异常：早晚高峰、车辆限行

对于常规性异常，因为其可能是由政策，经济或社会等因素影响而造成的日常结果，本模型不予报告异常.

突发性异常：由交通事故、大型活动等突发事件引起

对于突发性异常，我们认为这是交通管理部门可以实时解决的，因此将会报出异常.

3.2.2 数据定义

覆盖时间范围无间断城市出租车的 GPS 数据为研究交通异常提供了难得的资源,因为出租车在城市中的特殊性：每条道路都可能出现出租车，且可一定程度上代表普通车辆出现概率，因此我们以出租车的流量数据代表所有车辆流量数据进

行计算.

3.2.3 邻居道路定义

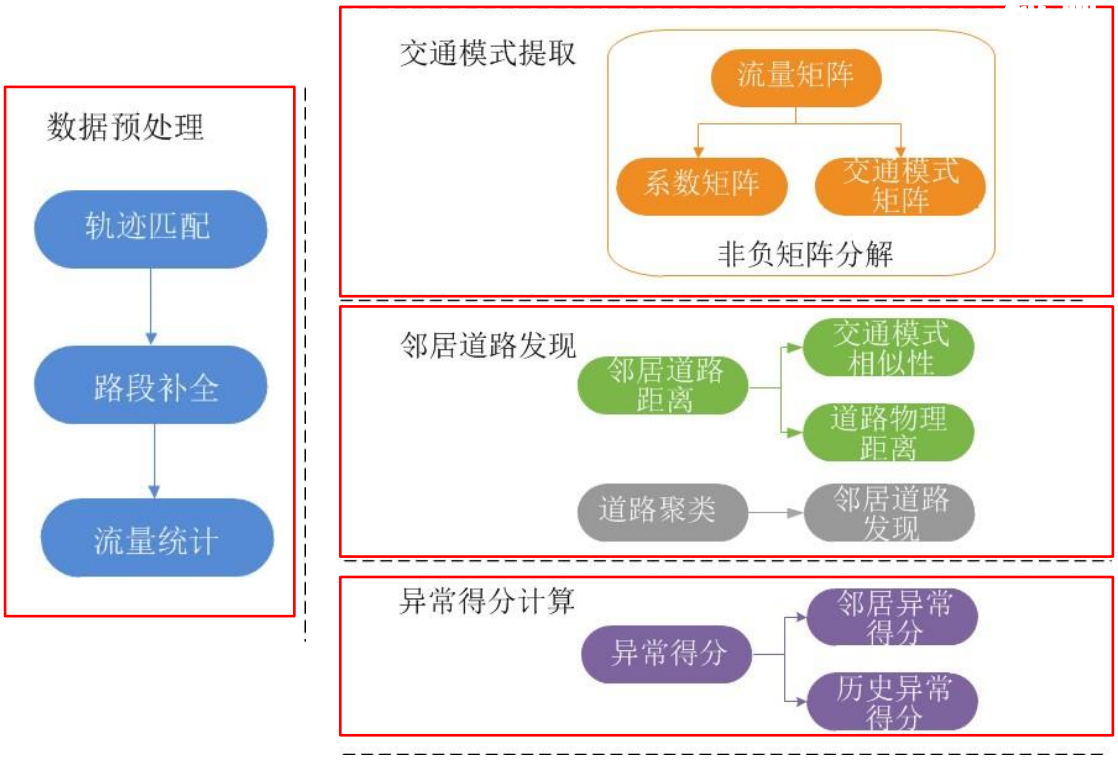
本模型从横(与实时同类道路相比)，纵(与历史自身数据相比)两方面进行道路流量对比以确定是否异常，有如下定义

同类道路:

地理距离临近

交通模式(即各个时间段流量数据)相似

3.3 大体思路

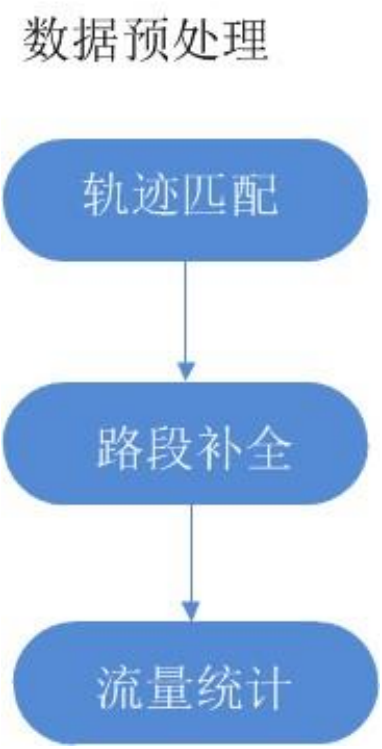


3.4 处理流程

3.4.1 数据预处理

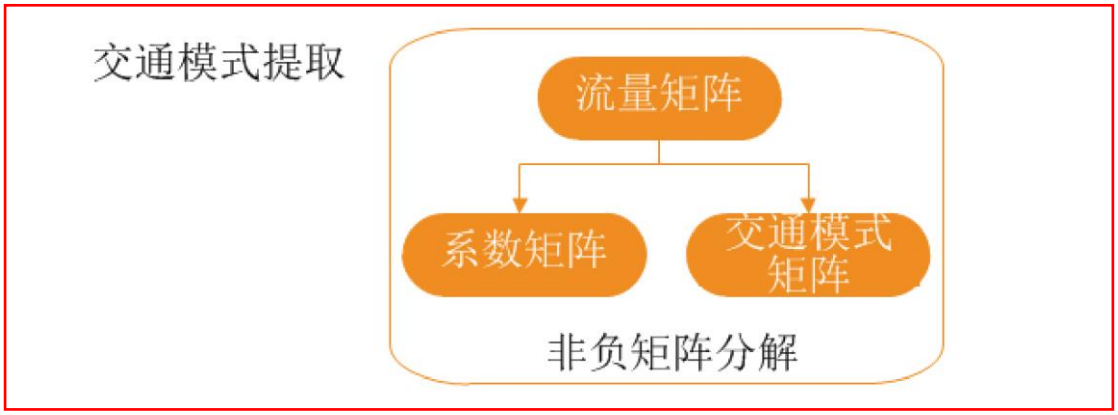
所获得的数据是不同出租车在全天各个时段(以 1min 为间隔)的位置，我们做如

下数据预处理:

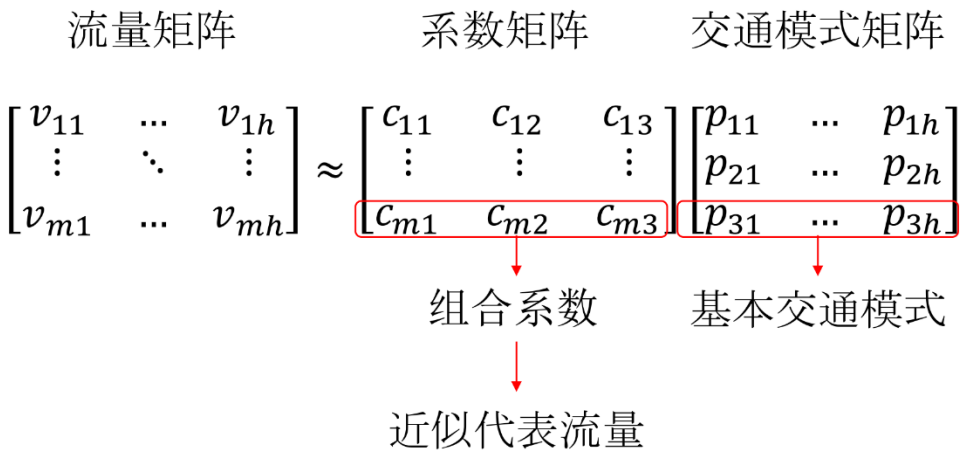


1. 使用 map-matching 算法将轨迹点匹配到道路上（OSM 地图数据）
2. 对于每条道路按照 15 分钟或 1 小时间隔计算各个时间段内的流量值
3. 剔除不合适数据，修补不合理数据
4. 总结数据，制成表格

3.4.2 交通模式提取，去除噪声

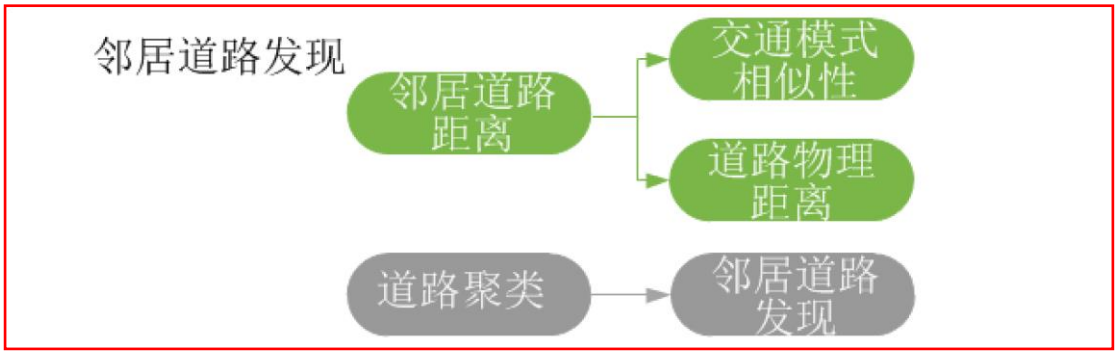


■ 非负矩阵分解



经过我们对于数据的预处理和查验，我们发现交通流量数据有很多噪声，例如交通异常情况很多，因此我们进行非负矩阵分解以去除噪声，同时提取出基矩阵(即交通流量模式矩阵)作为基本交通流量模式，提取出系数矩阵作为之后的聚类输入，这样可以有效的减少聚类时间，同时去除了大量噪声，有助于预测.

3.4.3 邻居道路发现



首先我们根据以下定义计算出邻居道路:

邻居：交通模式相似性、物理距离相近

模式相似性由 NMF 系数矩阵求得

物理距离通过道路经纬度信息求得

其次我们通过分析，选择以下公式定义道路距离:

$$D_{ij} = \alpha D_{模式_{ij}} + (1-\alpha)\, D_{距离_{ij}}$$

其中$D_{模式_{ij}}$代表了交通模式相似性，$D_{距离_{ij}}$代表了相邻道路的距离求得的相似性，α为线性组合参数.

最后我们根据所定义距离对道路聚类，同类道路互为邻居.

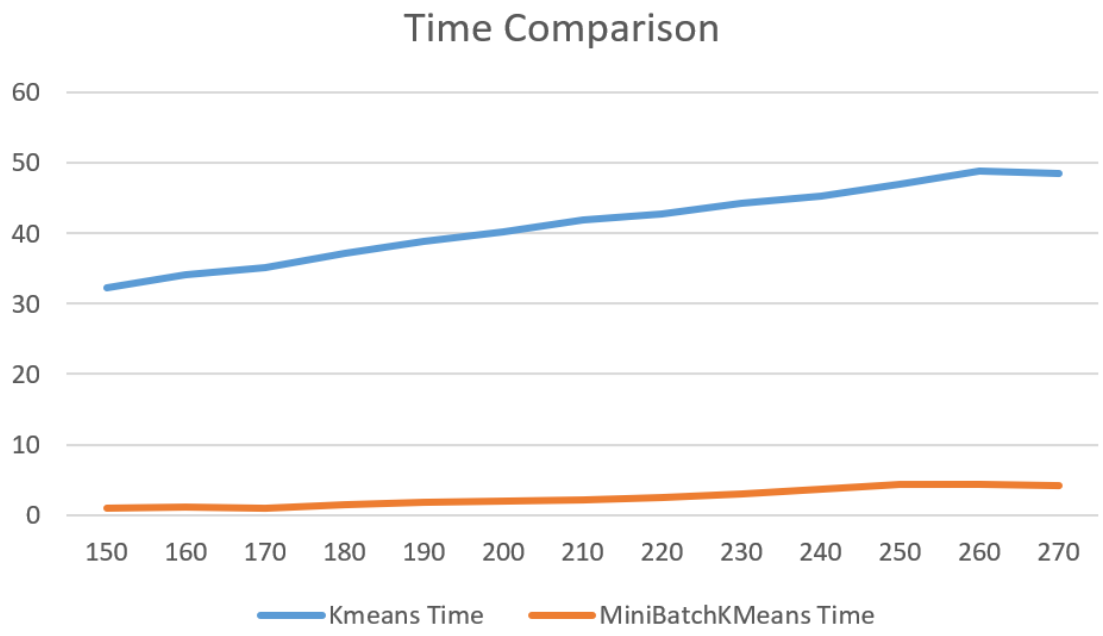
对于聚类算法的选择，我们首先对常用聚类算法 KMeans 和 DBSCAN 进行了对比，发现:

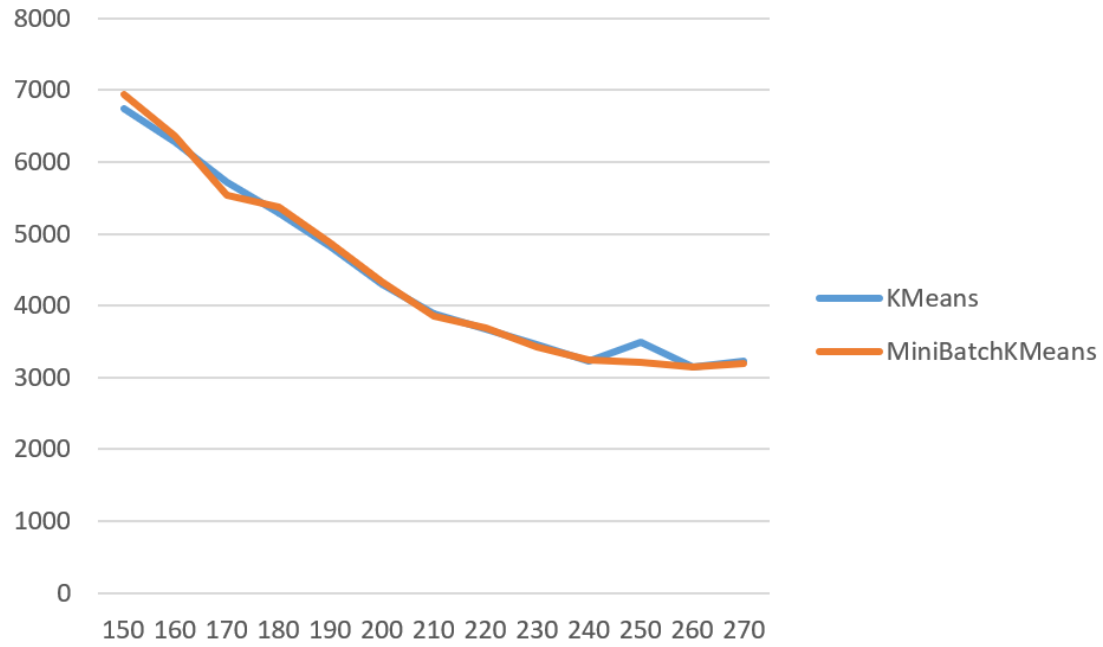
DBSCAN 聚类效果

```
Eps: 4.5e-06, minPts: 2
Cluster Number: 238
Noise Ratio: 25.2357284
Time: 8743.73652959
```

DBSCAN 聚类效果噪声很大，达 25%，同时时间很长，因此我们选择了几乎没有噪声点和时间很短的 KMeans 算法.

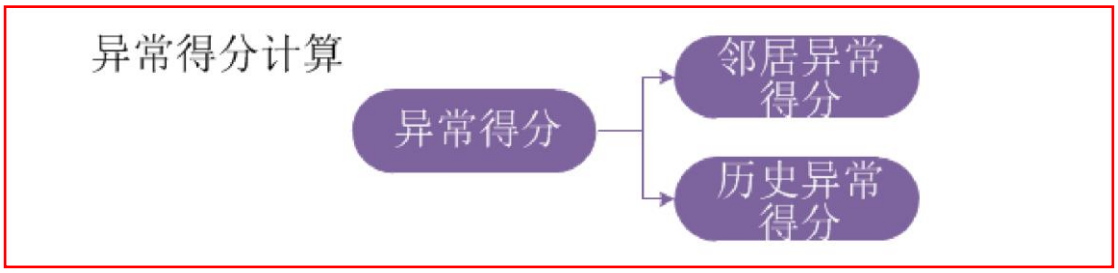
其次我们对不同 KMeans 算法(即传统 KMeans 算法和 MiniBatchKMeans 算法)进行了比较:





由此我们发现，在数据量很大的情况下，KMeans 与 MiniBatchKMeans 效果相差不大，但 MiniBatchKMeans 时间远小于 KMeans，因此我们选择了 MiniBatchKMeans 算法，根据上图亦可知，在聚类个数为 220 时，聚类时间与效果混合来看最好，因此我们选择了 220 类.

3.4.4 计算异常得分



邻居异常得分、历史异常得分

由概率密度函数求得

当前流量值在邻居或历史中出现的概率

概率越低，异常得分越低，越有可能是异常

对于分布函数的选择，我们对比了正态分布和泊松分布，考虑到正态分布可能会出现负值，我们采用了泊松分布.

得分公式:

$$S_{ij} = \beta S_{模式_{ij}} + (1-\beta)\frac{S_{距离_{ij}}}{\lambda}$$

其中，$S_{模式_{ij}}$代表与历史对比的得分，$\frac{S_{距离_{ij}}}{\lambda}$代表与邻居道路对比的得分，λ作为统一单位使用，$\beta$为线性组合参数.

我们定义，当$S_{ij} \leq \varphi$时，模型报告异常

3.4.5 模型结果

通过微博搜索的数据，我们得出了模型的一些预测实例，上图为老师展示实例，放在此处作为示范.

3.4.6 模型评估

我们采用了留出法对模型进行验证，训练集为 11.20-11.22 三天数据，预测集(检测集)为 11.26 数据，我们采用混淆矩阵进行模型评估，即:

真实数据	预测数据	
	预测异常	预测不异常
异常	预测准确(TP)	预测不准确(FN)

不异常	不准确预测(FP)	准确预测(TN)

评估的三个指标为:

P(准确率): $\mathbf{P} = \frac{TP}{TP+FP}$

R(查全率): $\mathbf{R} = \frac{TP}{TP+FN}$

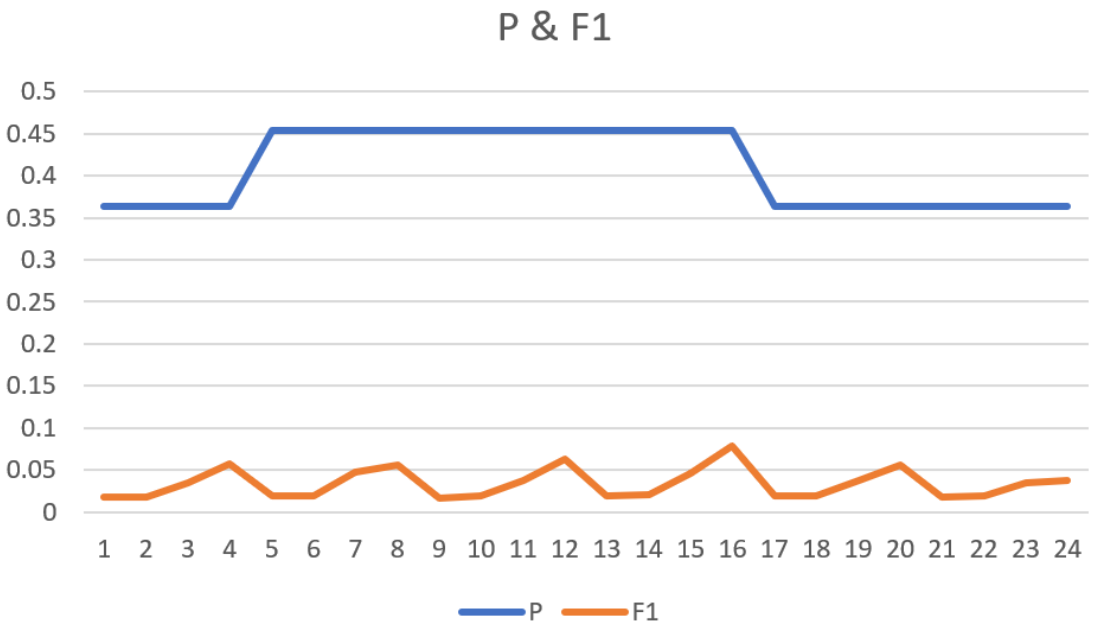
F1(调和平均): $\mathbf{F1} = \frac{2PR}{P+R}$

通过不断更改参数β与φ，我们得出如下结果:

β	φ	R	P	F1
0.38	0.001	0.0093536	0.363636	0.018238073
0.38	0.0001	0.0092423	0.363636	0.018026434
0.38	0.00001	0.01847362	0.363636	0.035160974
0.38	1E-06	0.0313	0.363636	0.05763874
0.43	0.001	0.0097735	0.454545	0.019135553
0.43	0.0001	0.0097429	0.454545	0.019076898
0.43	0.00001	0.0247362	0.454545	0.046919078
0.43	1E-06	0.0295	0.454545	0.05540426
0.48	0.001	0.0084437	0.454545	0.016579418
0.48	0.0001	0.0095632	0.454545	0.01873229
0.48	0.00001	0.019754	0.454545	0.037862538
0.48	1E-06	0.0335	0.454545	0.062401039
0.53	0.001	0.0098736	0.454545	0.019327372
0.53	0.0001	0.010248	0.454545	0.020044093
0.53	0.00001	0.024385	0.454545	0.046286847
0.53	1E-06	0.0434	0.454545	0.079234666
0.58	0.001	0.0096349	0.363636	0.018772406
0.58	0.0001	0.0099244	0.363636	0.019321476
0.58	0.00001	0.01952534	0.363636	0.037060715
0.58	1E-06	0.0302	0.363636	0.055768427

0.63	0.001	0.0093536	0.363636	0.018238073
0.63	0.0001	0.009465	0.363636	0.018449775
0.63	0.00001	0.0183246	0.363636	0.034890951
0.63	1E-06	0.0195	0.363636	0.037015065

转化为图像即:



由于数据不多，因此我们主要关心此模型的查准率和 F1 值，取其最优解，即为:

β	φ	R	P	F1
0.53	1E-06	0.0434	0.454545	0.079234666

此时我们认为模型效果最优，查准 5 次.

可以发现，可能由于 数据缺失 或是 预测集真实异常结果不准确 的原因，模型效果并不理想，有待改进.

4 项目总结

4.1 项目评价

虽然最后模型结果并不是太好，但项目的整个流程很值得学习与研究. 本项目中，我创造性的对比了各个算法和函数的优劣，通过一定的科学的评价标准，选出了

最优的方法，这是一个非常有意义的学习过程. 同时我顺利完成了整个项目的建立，推导，实现与评估，积累了很好的科研项目经验，对以后的学习很有帮助. 本项目的研究课题也很有意义，拥堵问题是世界性难题，我也因此开阔了视野，学习了一些解决问题的想法与思路. 同时本项目的研究方法较前人方法效果更为优秀，也更有理有据，是一个好的研究项目示范.

4.2 收获

参加科研项目的目的有二:

参与科研项目流程，学习数据挖掘方法，组织规划项目进度，亲自实践项目.

结交同学朋友，一起成长，共同进步.

在此次 7 天项目中，我亲身时间参与了项目流程，对于项目中不明白的问题也都通过问老师一一解答，还自己寻找了最优解的各个方法，以可视化的方式展现并分析. 同时项目进度规划也紧凑充实，加强了我的科研能力和小组沟通能力，收获颇多，结交很多好友.