贾成君 2014011552 自46

1. MDS算法是利用距离矩阵还原原始数据的算法，算法流程见书上内容。以下图片为自己手写和使用MATLAB的”mdscale”函数得到的MDS结果：



由于MDS结果是可以做旋转、镜像等线性变换的，两个结果其实大致相同（差别主要在于汉中与十堰、重庆形成的夹角）。

我们按照武汉-郑州-北京-周口-运城-十堰-汉中-重庆 的顺序在百度地图上标注出结果，如下图：



对比，可以发现：MDS结果和实际地图结果有一定的符合，但是在周口、运城等一些城市，其地理位置和实际地图差异较大。

这是由于我们的距离矩阵使用的是火车通勤时间——这个时间与实际地理距离不是呈现正比关系的。如：郑州到武汉的距离要比到周口的距离远，但是实际通勤时间却远小于到周口的时间。这个可能是由于选择的通勤时间受到火车车次的影响（K、T、Z、D、G列车的通勤时间是不一样的，如果都选择Z列车，效果可能会好一些）。

1. 选择PCA、t-SNE、LLE三种方法进行测试。
2. 首先，给出各个算法的说明和流程如下：

|  |  |
| --- | --- |
| 算法  说明 | PCA——  PCA是利用线性变换，使用正交向量表示原来的数据，使得投影后的数据方差和最大（保留信息最多） |
| 计算流程 | 输入：n\*p矩阵X、需要保留的位数k  计算：   1. 样本去中心D=X-EX 2. 协方差,进行奇异值分解 3. 取最大的k个特征值对应特征向量，构成投影阵W 4. 投影到   输出：n\*k矩阵X’(k<p) |
| 算法  说明 | t-SNE——  t-SNE利用概率分布度量样本之间的距离(j样本和i样本近邻概率)，使得投影后的数据对应的概率分布与原始数据误差最小 |
| 计算流程 | 输入：n\*p矩阵X、需要保留的位数k  计算：   1. 随机初始化投影后的向量X’（n\*k） 2. 根据分布之间差异程度(K-L距离)表达出损失函数[[1]](#footnote-1)，定义学习率 3. 使用梯度下降法进行迭代求解   输出：n\*k矩阵X’(k<p) |
| 算法说明 | LLE——  LLE是借助分段线性的思想，使得投影后的数据中，样本对应近邻的线性关系与原数据保持一致（误差最小）；其中近邻数目可以自行指定 |
| 计算流程 | 输入：p\*n矩阵X、需要保留的位数m、近邻数k  计算：   1. 对X中的每一个样本找到对应的k个近邻（距离最小的k个） 2. 使用这些近邻坐标，去掉样本坐标后，得到矩阵Z；计算对应的相关矩阵C=Z’\*Z 3. 解出C\*w=1,然后把w的维度扩展到n—非近邻的坐标对应置0，然后归一化。由此可以得到衡量线性相关程度的矩阵W 4. 利用M=(I-W)’\*(I-W)，找到M的最小的k个特征值，让Y的每一行对应特征向量[[2]](#footnote-2)   输出：n\*k矩阵Y(k<p) |

1. 把数据TrainX降低到二维，然后对不同的类使用不同的颜色进行表示；可得如下的降维结果：

|  |  |
| --- | --- |
| PCA |  |
| T-SNE[[3]](#footnote-3) |  |
| LLE[[4]](#footnote-4) |  |

投影之后的样本，能够明显看到：一类样本大致聚在一起（高维空间里它们是相近的），偶尔有一些交叉。

1. 由于降维算法需要先把数据进行去平均，不同的数据可能降维后的映射结果差异较大[[5]](#footnote-5)；所以我们把测试集和训练集合并，直接对比结果。

使用PCA压缩数据，发现：保留99%的信息量，需要296维数据，因此我们可以选择296作为压缩后的维度。

由于涉及多分类问题，分类器使用神经网络——隐藏节点数恒定取10；统计降维前后的分类[[6]](#footnote-6)错误率如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据 | 原始数据 | PCA-296维 | LLE-296维 | T-SNE-2维 | PCA-76维[[7]](#footnote-7) |
| 训练错误率 | 2.4% | 1.7% | 1.5% | 1.6% | 3.7% |
| 测试错误率 | 2.5% | 5.9% | 3.0% | 1.1% | 4% |

对比结果，可以发现：适当降维后（如果降维太多，可能损失信息过多，分类变得困难），训练错误率会有所下降，测试错误率会有所提升（直观理解，这是由于信息损失带来的影响）。

1. 对比各种降维算法带来的错误率，从上表我们可以发现，PCA降低到296维、LLE降低到296维、T-SNE降低到2维后，训练错误率变化较少——这说明此时降维对于特征的提取比较良好；

而测试错误率PCA最大，其次是LLE，T-SNE最小（比使用原始数据进行训练得到的结果更小），直观理解，PCA算法只是尽可能的保留样本信息；而LLE和T-SNE都考虑到了样本之间的相对位置，后者对于样本分类来说是更加重要的信息，相比于LLE，T-SNE则通过复杂的迭代求解最优表达方式，这种非线性的映射能使得样本之间的关键位置信息集中到很小的维度，从而提高分类效果。

对比，降低到不同维度的PCA对于分类的影响，我们可以明确：信息的集中对分类有所影响，但是信息损失不大的情况下，这种做法可以提高运算效率；我们需要在效率和信息的全面之间做权衡。

首先针对原来的数据——包含所有的特征——尝试进行分类，选择较好的分类器。我使用SVM进行分类，分别测试线性核、二次核、RBF核的分类效果，最后发现线性核的分类效果比较好[[8]](#footnote-8)。

使用线性核SVM、利用所有特征进行分类，并借助MATLAB集成的SVM分类器里的十倍交叉验证函数kfoldLoss进行十倍交叉验证；可得：平均错误率为9.25%。

然后，使用过滤法，单独使用单一特征进行分类，可得最小的前8个错误率为：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 特征编号 | 48 | 917 | 220 | 5 | 416 | 428 | 566 | 1 |
| 错误率 | 0.115 | 0.1325 | 0.2325 | 0.2525 | 0.2525 | 0.4450 | 0.4475 | 0.4500 |

所以，首先，我们直接依次使用48、48+917、48+917+220…进行错误率计算，交叉验证错误率结果如下：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 使用特征个数 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 错误率 | 0.115 | 0.09 | 0.06 | 0.0525 | 0.050 | 0.040 | 0.050 | 0.0550 |

从结果来看，使用前6个特征——5、48、220、416、428、917的组合，分类效果最好，此时对应的错误率约为4%；之后增加特征个数，错误率上升的原因可能是由于其他特征对于分类没有贡献，由于样本比较少，反而对于分类带来比较大的噪声影响。

使用包裹法进行特征选择，得到的结果依次为：

48、917、5、220、323、129、315、630、19、9，使用这些特征进行分类，可得：错误率

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 使用特征个数 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 错误率 | 0.115 | 0.0925 | 0.065 | 0.045 | 0.040 | 0.0275 | 0.030 | 0.0350 | 0.0250 | 0.030 |

可见，选择5、48、129、220、323、917的组合比较好。

分析选择过程，可知由于样本较少，特征较少的分类效果会比较多的特征可能更好，而由于特征的组合可能会带来比单个特征更好的效果，所以可以通过包裹法进行特征选取以取得更好的效果。

1. 更加数学化的表达参看：<http://lvdmaaten.github.io/publications/papers/JMLR_2008.pdf> [↑](#footnote-ref-1)
2. 更加数学化的表达参看：<https://www.cs.nyu.edu/~roweis/lle/algorithm.html> [↑](#footnote-ref-2)
3. 使用的代码，首先PCA降维到50，初始化随机使用的方差为30 [↑](#footnote-ref-3)
4. 近邻数取60 [↑](#footnote-ref-4)
5. PCA可以通过使用映射矩阵的方法，实现测试集转变到压缩后的结果；但是这一般要求测试集和训练集的样本均值相等，这样两者直接去均值后映射压缩矩阵可以近似认为一致——此次使用的集合，分别计算PCA压缩方法后对应的映射矩阵，发现相差较大——当使用训练集的压缩映射矩阵直接压缩测试集，信息丢失较大。其他两个方法也有类似问题。 [↑](#footnote-ref-5)
6. 由于T-SNE使用的是迭代计算进行压缩，计算量较大；因此使用该方法测试时，维度取2。 [↑](#footnote-ref-6)
7. 此时信息率达到90% [↑](#footnote-ref-7)
8. 直观考虑，样本的数量较少，而特征维度非常多；所以复杂的模型效果可能比较差。 [↑](#footnote-ref-8)