浅谈流形学习中的几种经典降维方法

这几年看的知识多，忘的也多，诸如Manifold Learning、Spectral Graph Theory等一些名词最早是研一在北大数院姚远老师开的一门数据分析课中接触到的，不过由于自己数学实在太渣，不仅学的时候云里雾里，而且期末一结束就基本对这些概念忘得七七八八了，这段时间正好在工作中遇到相关的问题，在此进行简单回顾。

（本文主要从一个计算机系学生的角度来介绍几种经典的流行学习方法，争取不过多的关注各种数学细节，而是从数据分析的角度进行理解。）

本文涉及的方法包括：MDS(Multidimensional Scaling)、ISOMAP、LLE(Locally Linear Embedding)、LE(Laplacian Eigenmaps)、Diffusion Map。

另外，pluskid大神有个聚类相关的系列文章写的非常棒，地址在这里。

1. 流形学习(Manifold Learning)基本概念

维基百科上给流形下的定义是

"In mathematics, a manifold is a topological space that locally resembles Euclidean space near each point. More precisely, each point of an n-dimensional manifold has a neighbourhood that is homeomorphic to the Euclidean space of dimension n."

这句话还是很好理解的，大致意思就是所谓的流形(Manifold)是一种局部可以近似为欧式空间的特殊拓扑空间，并且n维流形中的任意一个点都可以找到一个邻域，使得它们之间的结构关系可以近似嵌入到一个n维欧式空间中去。例如地球虽然在三维空间中（近似）是一个球体，但在我们这些生活在其表面上的人眼中却完全可以近似为一个平面，并且我们观察的区域越小，就越能够忽略地球表面曲率的影响。 不过仅仅有流形的一个概念性描述显然并没什么用，我们至少还需要某种能够确定流形中任一一个点的方法，或者用更加数学的语言来说，我们需要定义坐标系。不过该怎么定义呢？对于一些标准的几何体我们或许可以通过某些公式进行定义，比如球面既可以在xyz坐标轴下用满足

$$x^2 + y^2 + z^2 = R^2$$

的坐标

$$(x, y, z)$$

表示，也可以在极坐标下用

$$(\rho, \phi, \theta)$$

表示，但是我们知道很多复杂、不规则的流形至少目前是完全没办法用精确数学公式来描述的（例如经典的手写数字识别问题中所有图片构成的流形，每张图片都是一个

$$8 \times 8$$

的矩阵），这时又该怎么办呢？好吧，数学家的idea总是很充沛的，他们总结出两类坐标系来解决这个问题，就是分别在内蕴空间(

intrinsic space)和外围空间(ambient space)使用的坐标，前者是流形内部空间使用的坐标（例如地球表面的经纬度），后者是把流形嵌入到一个外围空间（例如欧式空间）以后各个点在外围空间中的坐标，一般来说如果我们需要分析的流形比较复杂，很难对其内蕴空间进行描述，那我们可以通过分析其外蕴空间下的坐标来对其内部结构进行理解。

流形学习里面可以研究的问题非常多，本文主要介绍的降维问题大体上都沿着一个框架进行解决，即**先通过某种方法对原流形内蕴空间进行逼近，并求取各个点间距离 / 相似度构成的矩阵，最后通过一个**

等距重构**(isometry)操作将所有数据映射到新的低维空间**，各种不同方法主要优化的还是前面计算距离 / 相似度矩阵部分，而后面的等距重构相对算是一个比较简单成熟的问题，下面我们将从最简单的线性情况开始讲述，逐渐深入到复杂的非线性情况，并通过各种不同的角度对距离 / 相似度进行理解建模。

1.5 MDS(Multidimensional Scaling)

讲道理MDS其实并不属于流形学习的范畴，完全就是一个线性算法，不过由于后面要说的ISOMAP和它关系非常紧密（换汤不换药的典型代表），这里也临时插入简单介绍一下。

MDS算法的所解决的核心问题是：给定一系列点 $X = [x_1, x_2, \cdots, x_n] \in \mathbb{R}^{p \times n}$ 在原空间中的相互距离 $D \in \mathbb{R}^{n \times n}$ ，求这些点在一个新的q维欧式空间中的坐标表示 $\tilde{X} \in \mathbb{R}^{q \times n}$ ，并使各个点之间的距离还能尽可能保持，即 $\min_{\tilde{X}} \sum_{i,j} (\|\tilde{x_i} - \tilde{x_j}\|_2^2 - d_{ij})^2$ 。从这里可以看出，MDS和PCA非常相似，都是要计算数据样本点在某个空间中的表示，差别仅在于PCA的输入是样本点在原空间的坐标，而MDS则需要输入各个点的距离度量，事实上已经有证明如果MDS输入的是样本点在原空间的欧式距离，则与PCA等价，其求解方法也是通过对一个由D变形得到的矩阵进行特征分解，然后取前q个特征值对应的特征向量以计算各点的q维坐标，具体数学推导如下，不想看的可以直接跳过。

为了方便公式推导，这里所说的距离矩阵D定义为距离平方的矩阵

$$D = (d_{ij}^2) = k \cdot \mathbf{1}^T + \mathbf{1} \cdot k^T - 2K$$

其中 $\mathbf{1} = (1, 1, \cdots, 1)^T, K = X^T X, k = diag(K_{ii}) \in \mathbb{R}^n$ 。定义数据均值和减去均值后的数据为

$$\hat{\mu} = \frac{1}{n} \cdot X \cdot \mathbf{1}$$

$$\tilde{X} = X - \frac{1}{n} X \cdot \mathbf{1} \cdot \mathbf{1}^T$$

那么

$$\tilde{K} = \tilde{X}^T \tilde{X} = K - \frac{1}{n} K \cdot \mathbf{1} \cdot \mathbf{1}^T - \frac{1}{n} \mathbf{1} \cdot \mathbf{1}^T \cdot K + \frac{1}{n^2} \mathbf{1} \cdot \mathbf{1}^T \cdot K \cdot \mathbf{1} \cdot \mathbf{1}^T$$

现在我们再定义一个神奇的矩阵B

$$B = -\frac{1}{2} H \cdot D \cdot H^T$$

其中 $H = I - \frac{1}{n} \cdot \mathbf{1} \cdot \mathbf{1}^T$ 。由于 $k \cdot \mathbf{1}^T \cdot H^T = k \cdot \mathbf{1}(I - \frac{1}{n} \mathbf{1} \cdot \mathbf{1}^T) = 0$ ，取转秩则有 $H \cdot \mathbf{1} \cdot k^T = 0$ ，带入B的表达式则有

$$B = H \cdot K \cdot H^T = K - \frac{1}{n} K \cdot \mathbf{1} \cdot \mathbf{1}^T - \frac{1}{n} \mathbf{1} \cdot \mathbf{1}^T \cdot K + \frac{1}{n^2} \mathbf{1} \cdot \mathbf{1}^T \cdot K \cdot \mathbf{1} \cdot \mathbf{1}^T = \tilde{K}$$

即

$$B = -\frac{1}{2} H \cdot D \cdot H^T = \tilde{X}^T \tilde{X}$$

如果看到这里还没晕掉的同学应该可以发现，这个公式意味着通过D和H然后加上特征值分解操作，我们就可以得到X在减去均值后的坐标表示了！如果特征值分解后向PCA那样只取前q个最大特征值对应的特征向量，就能获得我们想要的所有点的q维特征表示！

不知道别人学习到这个算法时有什么感受，我自己在第一次看到上面定义的B和H时整个人其实完全是懵逼状态，搞不清这两个矩阵是什么含义，比如为啥需要定义它们，为啥要这么定义它们，后来看到最后一步公式的时候恍然大悟，整个过程其实是在构造距离矩阵D和原始坐标的关系，也就是设计函数 $F(D)$ 和 $G(\tilde{X})$ 使得 $F(D) = G(\tilde{X})$ ，并且 $G(\tilde{X})$ 可以反向解出 $\tilde{X}$ ，虽然之后随着时间会对具体推导步骤逐渐遗忘，不过整个推导思路还是记忆深刻。

2. ISOMAP

讲完MDS终于可以开始介绍ISOMAP了，不过说起来其核心算法就一句话：“把原始由欧式距离计算得到的D矩阵换成流形学习中的测地距离(geodesic distance)，然后执行MDS”。WTF，就这么简单那这算法怎么还能这么出名。。。？答案是：没错，它们唯一的差别就是使用的距离矩阵。

事实上ISOMAP就是流形学习框架下的MDS，具体来说，原始MDS使用的欧式距离可以算是目标流形的外蕴空间距离，而测地距离则是目标流形的内蕴空间距离，理论上如果要把流形数据映射到一个新的欧式空间，那么内蕴空间距离肯定会比外蕴空间距离更加准确。不过等下，之前不是还说这种复杂流形的内蕴空间坐标系无法确定么，这个测地距离又该如何计算？请看ISOMAP的具体步骤：

构建图模型 $G = (V, E, d_{ij})$ ，构建边时可以选择complete graph, k-nearest neighbours, ϵ -neighbours等；
使用Dijkstra、Floyd等算法计算各个点间的最短路径距离，并更新 d_{ij} ；

执行标准MDS算法计算数据点的新坐标。

原来，这里的思路是使用数据点在图模型上的最短路径距离来逼近流形的测地距离，可以想象如果我们的训练样本足够多，那么这样计算得到的测地距离很可能会有非常高的精度（代价是耗时非常高）。不过虽然这个算法的核心思想非常简单，但如果去看看实际ISOMAP的代码，则可能会发现很多额外的细节操作（例如outlier detection、feature scaling等），最终试验结果也可能差距非常之大。

3. LLE(Locally Linear Embedding)

上面的ISOMAP希望各个样本在新空间中可以尽量保持它们在原流形内蕴空间中的距离，但是很多时候我们可能只需要让原来距离近的样本在新空间继续保持近的距离，而原来非常远的样本现在在哪我们并不关注，换句话说，我们只关注样本的局部分布，这个思路催生了另一种新的流形学习算法，LLE。

那么如何保持数据的局部分布呢？LLE的思路是：如果某些样本靠的足够接近，那么其中任一一个样本必然可以通过其它样本进行重构，即

$$x_i = \sum_{j \in \mathbb{N}_i} W_{ij} x_j,\ s.t. \sum_j W_{ij} = 1, \forall i \in \{1, 2, \cdots, n\}$$

其中 $\mathbb{N}_i$ 表示点i的所有相邻点。整个算法的优化目标为

$$\min_W \sum_i \|x_i - \sum_{j \in \mathbb{N}_i} W_{ij} x_j\|_2^2$$

实际求解可以通过拉格朗日法计算，这里略过。LLE相比ISOMAP有以下几点好处：

由于图在建立时只考虑临近点（一般使用knn），所以边的数量为O(kn)；
W矩阵非常稀疏，非零值的比例为 $kn/n^2 = k/n$ ；
ISOMAP无法处理nonconvex manifolds with holes，LLE则可以处理。

4. LE(Laplacian Eigenmaps)

LE(Laplacian Eigenmaps)这个方法和LLE非常相似，目标都是映射后尽量保持数据的局部分布而忽略全局分布，但在解决方案上二者却有一些微妙的差别，相比LLE那样需要估计出一个W矩阵，LE算法则更为简单直接的定义W矩阵为点与点之间的相似度矩阵，使用常见的Gaussian Heat Kernel则为

$$W_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{t}} & j \in \mathbb{N}_i \\ 0 & \text{otherwise} \end{cases}$$

然后计算unnormalized graph Laplacian

$$L = D - W$$

其中D是图模型的度矩阵， $D = diag(W \cdot \mathbf{1})$ 。

normalized graph Laplacian为

$$\mathcal{L} = D^{-\frac{1}{2}} L D^{\frac{1}{2}}$$

优化目标为

$$\min_{\tilde{X}^T D\tilde{X}=1, \tilde{X}^T D\mathbf{1}=0} \frac{1}{2}\sum_{i,j} W_{ij}\|\tilde{x}_i - \tilde{x}_j\|^2 = \min_{\tilde{X}^T D\tilde{X}=1, \tilde{X}^T D\mathbf{1}=0} \mathrm{trace}(\tilde{X}^T L\tilde{X})$$

其中约束条件一 $\tilde{X}^T D\tilde{X} = 1$ 的作用是为了保证新坐标的scale不能太小（论文原文是”The constraint removes an arbitrary scaling factor in the embedding”）；约束条件二 $\tilde{X}^T D\mathbf{1} = 0$ 则是为了防止落入把所有点都映射到相同坐标的trivial解，注意， $\mathbf{1}$ 是L关于特征值0对应的一个特征向量，对应图模型中至少存在的那一个连通域。

注意上面这个优化公式和PCA优化目标的相似与差别，PCA求的是最大值，这里求的是最小值，因此我们只需要把L进行特征分解，然后取q个最小的非零特征值所对应的特征向量组成 $q \times n$ 的矩阵，即为我们需要的新的空间坐标。

5. Diffusion Map

前面的ISOMAP、LLE、LE等方法的基本思路都是通过用数据点在欧式空间的局部分布来逼近原流形空间的局部分布，以此推测其全局结构，但是本节将要进行介绍的Diffusion Map则很不一样，它从概率或者说随机游走(Random Walk)的角度出发，用各个点之间的连通性来描述远近，例如假设点A与点B之间有很多条高概率的路径，之前几种方法几乎都是选择概率最大（距离最近）的一条来定义距离，而Diffusion Map则会把所有这些路径的概率值全部加起来，整个求解过程就像某个对象在整个图模型中进行随机游走，计算最终平稳后落在每个点的概率。

具体实现起来，第一步依然是建立 ϵ -neighbours graph或者k-nearest graph，然后定义相似度矩阵W，那么考虑随机游走的idea，定义

$$D = diag(W \cdot \mathbf{1})$$

$$P = D^{-1}W$$

P矩阵可以认为包含了目标对象从任意一点i经过一步转移到j的概率值，接下来如果我们继续计算P的k次幂，则可以得到从任意一点i经过k步跳转到j的概率值，可以想象，通过不断计算 P^t 我们可以知道整个数据在不同时刻的概率分布变化过程，一般把这个过程称为**Diffusion Process**。

有了上面的Diffusion Process，我们可以定义一个更一般性的距离矩阵，

$$D_t(X_i, X_j)^2 = \sum_{u \in X} \|p_t(X_i, u) - p_t(u, X_j)\|^2 = \sum_k \|P_{ik}^t - P_{kj}^t\|^2$$

之后就用类似MDS的方法，在新空间中计算一个最能保持这个距离的坐标即可。

话说第一次看到上面这个距离定义时我大概没怎么仔细想直接略过了，结果前两天再次看到这个公式时大脑立马懵逼，论文里对这个定义的解释是: “The diffusion distance is small if there are many high probability paths of length t between two points. Unlike isomap’ s approximation of the geodesic distance, the diffusion metric is robust to noise perturbation, as it sums over all possible paths of length t between points.”，看起来就是非常直接的把各条路径的概率求和，然后各条路径的概率应该是子路径的概率乘积（i经过k再到j的概率是 $P_{ij} = P_{ik}P_{kj}$ ），但是这里拿两个随机游走的概率值相减是什么鬼？想了一天没结论，最后受不了去求助研究生英语课认识的北大数院的成同学，问题终于得以解决，原来，这个距离完全不是这个意思，而是在比较i、j跳转到其它点u的概率分布差异性，例如i、j都有0.3的概率跳转到A，0.1的概率跳转到B，0.6的概率跳转到C，那么i和j的距离就应该是0（相似度高），相反，如果i到A、B、C的概率分别为0.3、0.1、0.6，而j到A、B、C的距离分别为0.3、0.6、0.1，那么i和j的距离应该很远（相似度低），从某种角度上看，Diffusion Map可以算是一种soft版本的LE，同样是局部点的分布影响较大，远的点影响较小（因为 P_{ik}, P_{kj} 太小的话，即使差异达到成百上千倍产生的差值也很小），不同在于LE的相似度矩阵W是用 ϵ -neighbours graph或者k-nearest neighbours graph产生的边，而Diffusion Map则通过t来调节扩散的scale。

关于Diffusion Map，维基百科里有段话概括的非常好，

“One of the main ideas of diffusion framework is that running the chain forward in time (taking larger and larger powers of P reveals the geometric structure of X at larger and larger scales (the diffusion process). Specifically, the notion of a cluster in the data set is quantified as a region in which the probability of escaping this region is low (within a certain time t). Therefore, t not only serves as a time parameter, but also has the dual role of scale parameter.”

从这里可以看出Diffusion Map中的cluster实际上很像黑洞，一旦进入就很难逃逸出来。

到这里已经把几种经典的非线性降维方法介绍完，而且近几年里除了上面这些其实还有很多更新、更好的方法在不断被提出（例如Hessian Eigenmaps、LTSA），然而非常可惜的是在我研一这门课结课时老师说了一句：“虽然Manifold Learning的理论非常漂亮，不过实际应用时一般复杂度太高，对噪声较为敏感，因此并没有什么实际应用价值。。。”，sad story。

2017-02-10 • Category: 计算机 • Tag:

---

nicklhy
Just for fun!