原 流形学习——局部线性嵌入算法LLE

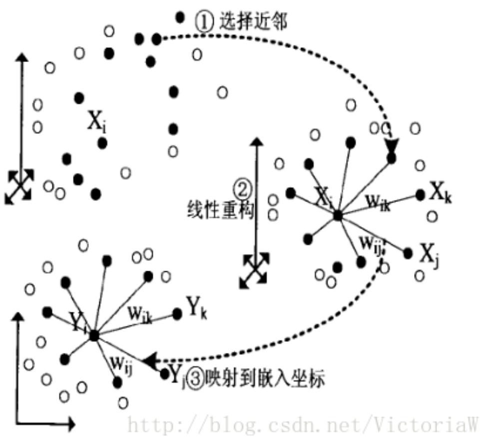
2017年11月10日 10:40:58 CodeTutor 阅读数：487 标签： 算法 降维 机器学习 LLE 更多

LLE原理

局部线性嵌入（Locally Linear Embedding, LLE）是无监督非线性降维算法，是流行学习的一种。

LLE和Isomap一样试图在降维过程中保持高维空间中的流形结构。Isomap把任意两个样本点之间的测地距离作为流形结构的特征，而LLE认为局部关系刻画了流形结构。

LLE认为，在高维中间中的任意一个样本点和它的邻居样本点近似位于一个超平面上，所以**该样本点可以通过其邻居样本点的线性组合重构出来。**



我们假设共有N个样本点。可以根据欧式距离或者其他相似性度量为每个样本点$x_i \in \mathbb{R}^d$找到K个邻居，用η_{i_k}表示x_i的第k个邻居点。重构误差为

$$J(W) = \sum_{i=1}^{N} ||x_i - \sum_{k=1}^{K} w_{ik}\eta_{i_k}||^2 \tag{1}$$

其中w_{ik}表示在重构x_i时的第k个邻居的权重系数。把所有的重构系数放在矩阵$W \in \mathbb{R}^{N\times K}$中，它的第$i$行元素表示重构$x_i$时的邻居系数。

为了得到W，求解最小化问题

$$\begin{aligned} \min_W \quad & J(W) \\ s.t. \quad & \sum_{k=1}^{K} w_{ik} = 1, i = 1, 2, \cdots, N. \end{aligned} \tag{2}$$

为了使得流形结构在低维空间中得以保持，LLE要求低维空间中的样本点仍能保持上面的局部线性关系。假设x_i在低维空间中的映射为$y_i \in \mathbb{R}^{d'}$，令$Y = [y_1, y_2, \cdots, y_N]$。$Y$可以通过下面的优化问题进行求解：

$$\begin{aligned} \min_Y \quad & \sum_{i=1}^{N} ||y_i - \sum_{j=1}^{N} w_{ij}y_j||^2 \\ s.t. \quad & \frac{1}{N}\sum_{i=1}^{N} y_i y_i^T = I. \end{aligned} \tag{3}$$

注意，这里的w_{ij}和前面的w_{ik}不完全一样，表示在低维空间中重构第i个样本点时，第j个样本点的权重。相应的矩阵$W \in \mathbb{R}^{N\times N}$可以由上面的$W$构造出来。为了不引起混淆，我们把后者重新写作$W'$。当$j$样本点是$i$样本点的邻居时，$w_{ij}$等于$W'$中对应的那个权重值；否则$w_{ij} = 0$。后面会统一称作$W$，根据上下文确定到底是哪个。

两个优化问题的求解

上面两个优化问题都可以直接得到最优解的解析式。

高维空间中的优化问题

方法一

令w_i表示矩阵W的第i行元素，$N_i = [\eta_{i1}, \eta_{i2}, \cdots, \eta_{iK}] \in \mathbb{R}^{d\times K}$，则

$$\begin{aligned} J(W) &= \sum_{i=1}^{N} ||x_i - N_i w_i^T||^2 \\ &= \sum_{i=1}^{N} (x_i - N_i w_i^T)^T (x_i - N_i w_i^T) \\ &= \sum_{i=1}^{N} (x_i^T x_i - 2x_i^T N_i w_i^T + w_i N_i^T N_i w_i^T) \end{aligned} .$$

由于第一项和W无关，所以目标函数等价于

$$J(W) = \sum_{i=1}^{N} (-2x_i^T N_i w_i^T + w_i N_i^T N_i w_i^T).$$

构建拉格朗日函数

$$L(W, \lambda) = \sum_{i=1}^{N} (-2x_i^T N_i w_i^T + w_i N_i^T N_i w_i^T) + \sum_{i=1}^{N} \lambda_i (w_i 1 - 1),$$

求导得到：

$$\frac{\partial L}{\partial w_i} = -2x_i^T N_i + 2w_i N_i^T N_i + \lambda_i 1^T = 0, \tag{5}$$

$$\frac{\partial L}{\partial \lambda_i} = (w_i 1 - 1) = 0. \tag{6}$$

令

$$C^i = N_i^T N_i \in \mathbb{R}^{K\times K}. \tag{7}$$

由公式(5)可以得到

$$w_i = (x_i^T N_i - \frac{1}{2}\lambda_i 1^T)(C^i)^{-1}, \tag{8}$$

于是有

$$\begin{aligned} w_i 1 &= (x_i^T N_i - \frac{1}{2}\lambda_i 1^T) \begin{pmatrix} \sum\_m = 1^K (C^i)^{-1}\_1m \\ \sum\_m = 1^K (C^i)^{-1}\_2m \\ \vdots \\ \sum\_m = 1^K (C^i)^{-1}\_Km \end{pmatrix} \\ &= x_i^T N_i \begin{pmatrix} \sum\_m = 1^K (C^i)^{-1}\_1m \\ \sum\_m = 1^K (C^i)^{-1}\_2m \\ \vdots \\ \sum\_m = 1^K (C^i)^{-1}\_Km \end{pmatrix} - \frac{1}{2}\lambda_i 1^T \begin{pmatrix} \sum\_m = 1^K (C^i)^{-1}\_1m \\ \sum\_m = 1^K (C^i)^{-1}\_2m \\ \vdots \\ \sum\_m = 1^K (C^i)^{-1}\_Km \end{pmatrix} \\ &= [x_i^T \eta_{i1}, x_i^T \eta_{i2}, \cdots, x_i^T \eta_{iK}] \begin{pmatrix} \sum\_m = 1^K (C^i)^{-1}\_1m \\ \sum\_m = 1^K (C^i)^{-1}\_2m \\ \vdots \\ \sum\_m = 1^K (C^i)^{-1}\_Km \end{pmatrix} - \frac{1}{2}\lambda_i \sum_{n=1}^{K} \sum\_m = 1^K (C^i)^{-1}\_nm \\ &= \sum_{n=1}^{K} x_i^T \eta_{in} \sum\_m = 1^K (C^i)^{-1}\_nm - \frac{1}{2}\lambda_i \sum_{n=1}^{K} \sum\_m = 1^K (C^i)^{-1}\_nm \\ &= \sum_{n=1}^{K} \sum\_m = 1^K x_i^T \eta_{in} (C^i)^{-1}\_nm - \frac{1}{2}\lambda_i \sum_{n=1}^{K} \sum\_m = 1^K (C^i)^{-1}\_nm \end{aligned} ,$$

代入(6)中，得到

所以

$$\lambda_i = \frac{2(\sum_{n=1}^K \sum\_m = 1^K x_i^T \eta_{in} (C^i)^{-1}\_nm - 1)}{\sum_{n=1}^K \sum\_m = 1^K (C^i)^{-1}\_nm},$$

代入公式(7)中得到

$$w_i = \left(x_i^T N_i - \frac{(1 - \sum_{n=1}^K \sum\_m = 1^K x_i^T \eta_{in} (C^i)^{-1}\_nm)}{\sum_{n=1}^K \sum\_m = 1^K (C^i)^{-1}\_nm} 1^T \right) (C^i)^{-1}.$$

根据公式(9)得到所有的w_i，组成矩阵W。

方法二

$$\begin{aligned} J(W) &= \sum_{i=1}^N ||x_i - \sum_{k=1}^K w_{ik} \eta_{ik}||^2 \\ &= \sum_{i=1}^N || \sum_{k=1}^K w_{ik} x_i - \sum_{k=1}^K w_{ik} \eta_{ik}||^2 \\ &= \sum_{i=1}^N || \sum_{k=1}^K w_{ik} (x_i - \eta_{ik})||^2 \\ &= \sum_{i=1}^N ||(X_i - N_i) w_i^T||^2 \\ &= \sum_{i=1}^N ((X_i - N_i) w_i^T)^T ((X_i - N_i) w_i^T) \\ &= \sum_{i=1}^N w_i (X_i - N_i)^T (X_i - N_i) w_i^T \end{aligned} , \tag{10}$$

其中$X_i = [x_i, \cdots, x_i] \in \mathbb{R}^{d \times K}$。

构建拉格朗日函数

$$L(W, \lambda) = \sum_{i=1}^N w_i (X_i - N_i)^T (X_i - N_i) w_i^T + \sum_{i=1}^N \lambda_i (w_i 1 - 1),$$

求导得到

$$\frac{\partial L}{\partial w_i} = 2 w_i (X_i - N_i)^T (X_i - N_i) + \lambda_i 1^T = 0, \tag{11}$$

$$\frac{\partial L}{\partial \lambda_i} = (w_i 1 - 1) = 0. \tag{12}$$

令

$$S = (X_i - N_i)^T (X_i - N_i). \tag{13}$$

可以得到

$$w_i = -\frac{1}{2} \lambda_i 1^T S^{-1}, \tag{14}$$

代入公式(12)中，得到

$$\frac{1}{2} \lambda_i 1^T S^{-1} 1 - 1 = 0,$$

$1^T S^{-1} 1$是一个实数，所以得到

$$\lambda_i = \frac{2}{1^T S^{-1} 1},$$

代入公式(14)得到

$$w_i = -\frac{1^T S^{-1}}{1^T S^{-1} 1}. \tag{15}$$

暂时还没有证明(9)和(15)是否等价，留作以后的习题吧。

低维空间中的优化问题

令$Y = [y_1, y_2, \cdots, y_N] \in \mathbb{R}^{N \times N}$，$w_i$表示$W$的第$i$行元素。优化问题(3)的目标函数可以化简成：

$$
\begin{aligned}
J(Y) &= \sum_{i=1}^{N} ||y_i - \sum_{j=1}^{N} w_{ij} y_j||^2 \\
&= \sum_{i=1}^{N} ||y_i - Y w_i^T||^2 \\
&= \sum_{i=1}^{N} (y_i - Y w_i^T)^T (y_i - Y w_i^T) \\
&= \sum_{i=1}^{N} (y_i^T y_i - 2 y_i^T Y w_i^T + w_i Y^T Y w_i^T) \\
&= \sum_{i=1}^{N} y_i^T y_i - 2 \sum_{i=1}^{N} y_i^T Y w_i^T + \sum_{i=1}^{N} w_i Y^T Y w_i^T \\
&= \sum_{i=1}^{N} y_i^T y_i - 2 \sum_{i=1}^{N} [y_i^T y_1, y_i^T y_2, \cdots, y_i^T y_N] \begin{pmatrix} w_{i1} \\ w_{i2} \\ \vdots \\ w_{iN} \end{pmatrix} \\
&\quad + \sum_{i=1}^{N} (\, w_{i1}, w_{i2}, \cdots, w_{iN} \,) \begin{pmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_N^T \end{pmatrix} (\, y_1, y_2, \cdots, y_N \,) \begin{pmatrix} w_{i1} \\ w_{i2} \\ \vdots \\ w_{iN} \end{pmatrix} \\
&= \sum_{i=1}^{N} y_i^T y_i - 2 \sum_{i=1}^{N} \sum_{j=1}^{N} y_i^T y_j w_{ij} + \sum_{i=1}^{N} \sum_{m=1}^{N} w_{im} y_m^T \sum_{n=1}^{N} w_{in} y_n \\
&= \sum_{i=1}^{N} y_i^T y_i - 2 \sum_{i=1}^{N} \sum_{j=1}^{N} y_i^T y_j w_{ij} + \sum_{i=1}^{N} \sum_{m=1}^{N} \sum_{n=1}^{N} w_{im} w_{in} y_m^T y_n \\
&= \sum_{i=1}^{N} y_i^T y_i - 2 \sum_{i=1}^{N} \sum_{j=1}^{N} y_i^T y_j w_{ij} + \sum_{k=1}^{N} \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ki} w_{kj} y_i^T y_j \\
&= \sum_{i=1}^{N} y_i^T y_i - \sum_{i=1}^{N} \sum_{j=1}^{N} 2 w_{ij} y_i^T y_j + \sum_{i=1}^{N} \sum_{j=1}^{N} \left(\sum_{k=1}^{N} w_{ki} w_{kj} \right) y_i^T y_j
\end{aligned}, \tag{16}
$$

令

$$
\delta_{ij} = \begin{cases} 1, & if \quad i = j \\ 0, & if \quad i \neq j \end{cases},
$$

令

$$
M_{ij} = \delta_{ij} - 2 w_{ij} + \sum_{k=1}^{N} w_{ki} w_{kj}, \tag{17}
$$

则(16)可以写成

$$
J(Y) = \sum_{i=1}^{N} \sum_{j=1}^{N} M_{ij} y_i^T y_j. \tag{18}
$$

通过展开进行矩阵相乘，可以证明

$$
J(Y) = \sum_{i=1}^{N} \sum_{j=1}^{N} M_{ij} y_i^T y_j = tr(Y M Y^T).
$$

令$Z = Y^T$，写作$Z = [z_1, z_2, \cdots, z_N]$，那么

$$
J(Z) = tr(Z^T M Z). \tag{19}
$$

优化问题(3)的约束条件等价于

用Z表示的话为

$$\frac{1}{N}Z^TZ=I.$$

于是优化问题(3)现在变成

$$\min_{Z} \quad tr(Z^TMZ),$$

$$s.t. \quad \frac{1}{N}Z^TZ=I$$

拉格朗日乘子法，可以得到这个优化问题的最优解满足

$$Mz_i=\lambda_i z_i,$$

即最优解肯定是M的特征向量，但是我们只需要d'个特征向量，是哪d'个呢？先把(22)代入目标函数中，得到

$$J(Z)=tr(Z^TMZ)=\sum_{i=1}^{N} z_i^T M z_i=\sum_{i=1}^{N}\lambda_i z_i^T z_i.$$

又因为约束条件$\frac{1}{N}Z^TZ=I$其实等价于

$$\begin{matrix} z_i^T z_i=N, i=1,2,\cdots,N \\ z_i^T z_j=0, i\neq j \end{matrix},$$

所以

$$J(Z)=N^2\sum_{i=1}^{N}\lambda_i,$$

所以为了最小化$J(Z)$，应该取矩阵M的前d'个最小非零特征值对应的特征向量作为Z的列向量，得到Z，其转置即为Y。

综上，M的前d'个最小非零特征值对应的特征向量作为Y的行向量就得到最优Y。

LLE算法总结

算法流程

步骤一
首先根据欧氏距离或者其他度量标准得到每个样本的K个近邻，得到近邻矩阵N_i。

步骤二
根据公式(7)求出C^i，并且求逆矩阵$(C^i)^{-1}$。代入(9)中，得到根据每个样本的重构系数。

步骤三
把步骤二中的权重系数重新构建成稀疏矩阵W。根据公式(17)计算M，计算M的特征之和特征向量，取最小非零特征值对应的特征向量作为Y的行向量得到Y。

算法优缺点

优点

- 算法中只涉及矩阵运算，容易实现；
- 低维空间维度变化时，不需要重新运行LLE，只要在原有低维空间的基础上增加或者减去维度；

缺点

- 数据流形不能是闭合结构，否则LLE不再适用

疑问

1. 优化问题(18)怎么应用拉格朗日乘子法？
   自己尝试推导了一下，发现得到的方程组非常繁琐，不知道怎么化简。

参考

[1] Nonlinear dimensionality reduction by locally linear embedding. Sam T. Roweis and Lawrence K. Saul. 2000.
[2] 降维打击之LLE算法
[3] 机器学习降维算法三：LLE (Locally Linear Embedding) 局部线性嵌入

局部线性嵌入(LLE)算法解释
3756
局部线性嵌入(LLE)算法 给定N个输入向量{x1,x2,...,xN}，xRd，通过LLE算法，得到输出向量yi，i[1,N]，yi Rm，m (... 来自： alaclp的专栏

机器学习降维算法三：LLE (Locally Linear Embedding) 局部线性嵌入
1.1万
如引用请务必注明此文出自：http://blog.csdn.net/xbinworld LLE Locally linear embedding（LLE）[1] 是一种非线... 来自： Bin 的专栏

流行学习常用算法
2591
Isomap：等距映射。前提假设为低维空间中的欧式距离等于高维空间中的侧地线距离，当然该算法具体实施时是高... 来自： qq_18343569的博客

LLE(局部线性嵌入)
637
*基本思想 LLE的基本思想是在任意一个样本点和他的近邻点之间构造一个局部线性平面，这个样本点可以由他的领... 来自： PING_ASI的博客

LLE局部线性嵌入算法
6257
LLE局部线性嵌入算法 导语：很久没发博文了，今日抽个小空，整理下上个学期做过的东西，写成博文，供给初学者... 来自： Eleven-Seven工作小空间

流形学习——Isomap算法
2910
Isomap算法介绍Isomap（Isometric Feature Mapping）是流行学习的一种，用于非线性数据降维，是一种无监督算... 来自： CodeTutor

流形学习
491
经典流形学习： 1、MDS 2、ISOMAP 3、LLE 4、SpecturalEmbeding 6、SNE系列 MDS是在降维后保持样本之间的... 来自： zhangweiguo_717的博客

流形学习(Manifold Learning)理解
1129
流形不是一种形状，而是一种空间。专业点讲，一个d维的流形是一个其内任意点局部同胚于欧式空间RdR^d的d维... 来自： Blateyang的博客

matlab 流形学习LLE算法
下载 09-12
matlab 降维算法 图像处理识别 流形学习算法 LLE ...lle算法详解及matlab代码实现 立即下载 7积分/C币 ...精选26个Python实用技巧,想秀技能先Get这份技...

一句话总结LLE（流形学习）
25
一句话总结LLE（流形学习） 核心：用一个样本点的邻居的线性组合近似重构这个样本，将样本投影到低维空间中... 来自： 喜欢打酱油的老鸟

文章热词 动脑学院 机器学习 机器学习中的分层训练 机器学习中的凸优化 机器学习识别语句 机器学习中的特征计算

相关热词 流形学习 mds流形学习 流形学习框架 dae流形学习 多流形学习

博主推荐
换一批

九日王朝 关注 185篇文章
大饼博士X 关注 93篇文章
zzu小陆 关注 246篇文章

史上最直白的logistic regression教程 之 一
1.4万
Logistic Regession是什么Logistic Regression是线性回归，但最终是用作分类器。 为什么叫Logistic呢？因为它使用了... 来自： 大数据和机器学习研究

graph Laplacian 拉普拉斯矩阵
1.2万
拉普拉斯矩阵是个非常巧妙的东西，它是描述图的一种矩阵，在降维，分类，聚类等机器学习的领域有很广泛的应... 来自： zht的专栏

概率统计与机器学习：独立同分布，极大似然估计，线性最小二乘回归
2071
独立同分布独立性 概念：事件A，B发生互不影响 公式：P(XY)=P(X)P(Y)P(XY)=P(X)P(Y)， 即事件的概率等于各自事... 来自： Kelisiya

CodeTutor
关注
向TA提问

原创	粉丝	喜欢	评论
340	411	103	153

等级： 博客 7 访问： 74万+
积分： 1万+ 排名： 2418

最新文章

强化学习资料汇总

自动驾驶学习资料

Ubuntu 16.04上安装OpenVPN服务器（一）

Ubuntu 16.04上安装OpenVPN服务器（二）

在Sublime Text中使用Markdown

个人分类

Projects	2篇
前沿转载	1篇
人工智能	6篇
机器学习	69篇
深度学习	32篇

展开

归档

2018年6月	1篇
2018年5月	9篇
2018年4月	1篇
2017年12月	11篇
2017年11月	15篇

展开