**Week14**

Done：Data-driven KNN, IRLS, 常用优化方法，LPP,

To do:奇异值分解，牛顿法二阶收敛性证明，软阈值收缩及实现，协方差矩阵，海森矩阵与多元函数极值，

**Data-driven KNN**

**摘要**：

这篇文章研究了数据驱动的参数k的计算，确定了在KNN应用中的对不同样本的不同参数k，比如分类，回归，缺失值填充。通过重构一个测试样本和训练数据之间的稀疏系数矩阵来实现这点。在重构过程中，范数正则化用来生成element-wise的稀疏系数矩阵，Locally Preserving Projection（**局部保留投影**）用来维持数据的局部结构从而达到高效率。

**介绍：**

KNN是非参的基于实例的懒惰的算法。思想是相似的样本属于同一类的概率比较高。

KNN对一个查询找到它最近的k个邻居，然后用多数表决法预测它的类别。

k值的选取一直很有挑战性。有人提出贝叶斯方法，有人证明了固定的k值对大多数样本不适用。本文通过重构一个测试样本和训练数据之间的稀疏系数矩阵来确定K值，通过这个矩阵，可以获得针对每个样本的最佳k值，称为s-KNN。在重构中，用最小二乘损失函数来最小化重构误差，用一个范数正则项来产生element-wise稀疏性，从而对不同的样本生成不同的k值。还应用了LPP来**保留重构过程中数据的局部结构**，旨在提高重构的表现。与以前的KNN相比，本文有以下几点贡献：

1. 之前的KNN中k值固定，s-KNN中不同样本k值不同；
2. 不同于LASSO，本算法考虑了样本的局部结构；
3. 提出了新的优化方法来求解目标函数。

**相关工作：**

分类：Liu设计了一个在KNN框架下去除异常值的算法，采用了**相互近邻**，优点是可以识别出伪近邻。Weinberger用半定规划来学习**马氏距离**的度量，其中来自不同类别的样本被一个大间隔分离开。Goldberger提出了学习二次距离度量的方法，这个方法关注让已经学习到的距离保持低秩，从而节省内存和搜索消耗。

回归：

缺失值填充：Zhang等人提出了基于Grey的距离来取代传统KNN中的欧氏距离（GBKII），可以提高KNN填充算法的表现。Meesad提出了一个在微阵列中填充缺失值的方法，结合了基于KNN的特征选择和基于KNN的填充，然后用KNN算法估计缺失值。最近，Zhang提出了基于核函数的缺失值填充算法，将属性关联到数据，得到最优的统计参数：缺失值填充之后的均值和分布函数。

以上可以看出，现有的KNN算法对全体问题空间使用固定k值，经常导致低效的预测。

**提出的方法：**

**重构和LPP：**

给出，其中n,m,d分别是训练样本，测试样本和属性维度。为了获得重构系数矩阵，我们用训练样本重构测试样本，目标函数定义为：

 （1）

其中用来衡量和第个训练样本的相关性。

为了实现重构，用LPP来获得最佳的线性变换W。LPP能在新的空间里保持原始数据的局部结构，也就是W将高维数据转换到低维数据：

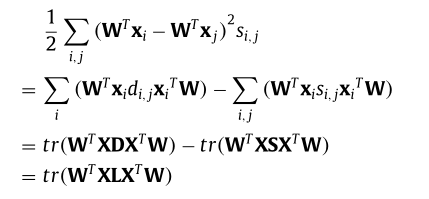
 （2）

于是LPP的目标函数可以定义为：

 （3）

其中S是权值矩阵，其中元素定义为，是可调参数。不失一般性，我们设。

联立（2）（3），以及一些运算，可以得到：



证明方法：

其中D是对角矩阵，第i个元素为。因此是**拉普拉斯矩阵**（图的度矩阵减去邻接矩阵）。性质：对称半正定；最小特征值为0；有n个非负实特征值。

**方法：**

我们用最小二乘损失函数控制重构误差：

 （5）

其中是X在Y的空间里的新的表示。这是凸函数，容易得到全局最优解。但是**不一定可逆**，因此加入范数：

 （6）

1. 式的最优解为闭式解.

范数被证明有稀疏性。在这篇文章里，我们希望**每个测试样本只由部分训练样本表示**，也就是说在W的每一列中有很多零元素。因此我们用范数代替范数，而且用LPP来保持数据在重构过程之后的局部结构。于是我们定义了s-KNN方法的目标函数如下：

 （7）

其中是可调参数，**用来平衡和的大小**。时，（7）式收缩到LASSO回归。

不同于LASSO，s-KNN考虑了通过LPP保持数据的局部结构。

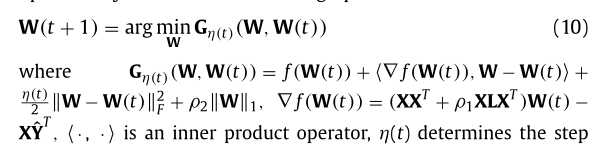
**优化：**

1. 式是凸但不平滑的函数，我们设计新的加速近端梯度算法解决这个问题。首先对（7）式应用近端梯度法，设：

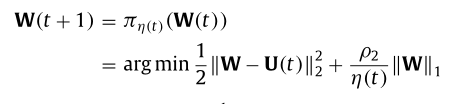
1543995720(1)

1543995738(1)

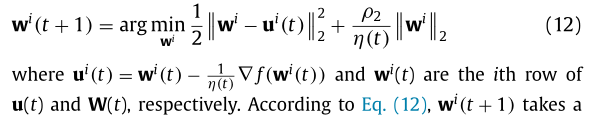
我们知道凸且可微（因为拉普拉斯矩阵L半正定），所以用近端梯度算法优化W，并用以下优化规则迭代它。



忽略与W无关的项，（10）可重写为：

****

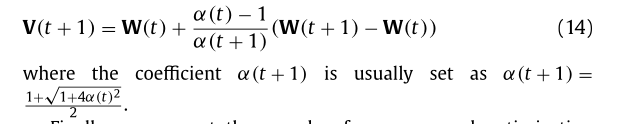
考虑到在每行上的可分性，我们分别为每行更新权值，则有：



这就变成了**软阈值收缩**的形式，（12）有闭式解：

**1544104039(1)**

同时，为了**加速近端梯度算法**，我们引入辅助变量：



最后我们证明算法的收敛性。

**结论：**

我们提出了新的s-KNN算法，对所有测试样本，基于数据的分布，用学习到的不同的k值替换固定k值。这是实例驱动的k值计算。关键是对测试样本和训练样本之间关系的重构，得到一个稀疏矩阵。由这个稀疏矩阵，对每个测试样本我们可以获得最佳k值。为了高效，用LPP保持数据的局部结构。

**LPP（局部保留投影）**

PCA和LDA是为了保持全局结构，而在许多现实应用中，局部结构更加重要。因此有LPP。

LPP：给出一个n维的数据集，寻找一个转换矩阵将这个点映射到维空间的个点：，（）也就是用表示，。

**算法步骤：**

1. 构建邻接图：G
2. 选择权值：
   1. 高斯核：
   2. 简单定义：如果邻接则
3. 特征映射：解这个方程：

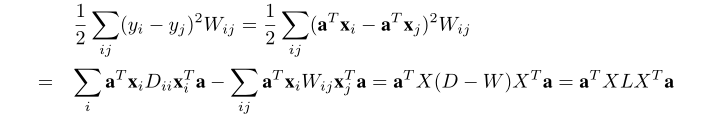
 （1）

其中D是对角矩阵，，是拉普拉斯矩阵。

**验证：**要最小化目标函数：



通过运算得到：



越大，则越重要。因此施加一个约束条件：



然后用拉格朗日法求解。

**IRLS**

**Q：为什么要加权，权值如何确定。**

1. **估计**

解线性方程组Ax =b时（A为, b为），当且仅当A为非奇异矩阵（方阵且满秩），方程组有唯一的精确的解。否则需要用估计方法求解。

如果b不在A的范围空间内（b不能用A的列向量线性表示），则方程组没有精确解。于是可以通过最小化误差向量的范数来求解估计问题。

1. **最小二乘误差估计**

如果上式没有唯一解，再考虑最小化x的范数，这样的组合问题通常有唯一解。

范数的最小化有解析解。均方误差定义为：。当A行满秩或列满秩时，此式可以以精确或近似的解来最小化结果。三种情况分别为：

1. 如果M=N, A为非奇异，则精确解为；
2. 如果M>N, 且A列满秩，则近似解为；
3. 如果M<N, 且A行满秩，则近似解为。

以上三个公式假设A行满秩或列满秩，如果不符合，可以用广义逆求解。

1. **加权最小二乘误差估计**

除了有解析解的情况之外，我们可以抛出一个更一般的问题，用加权的范数求最优的近似解来**强调方程的某些部分**。这里我们最小化。

其中W是权值的对角矩阵，近似解为：

1. 如果M>N, 且A列满秩，则近似解为；
2. 如果M<N, 且A行满秩，则近似解为.

以上是接下来IRLS的基础。

1. **其它范数的估计**

通常我们讨论范数，但是范数也有重要的用途。更一般的范数很灵活。向量的范数定义为：。

1. **范数估计**

IRLS算法是一个迭代的算法，用迭代重加权从加权最小二乘的解析解收敛到最佳的估计。

* 1. **方程大于未知数的超定系统**

误差向量：

 （1）

用（2）来求加权均方误差的最小值：

 （2）

* 1. **IRLS**

如果用估计来求解超定方程组，就是定义以下误差范数：， 并寻找x来最小化这个误差的p范数。

**已经证明这等价于求解一个有适当权值的最小加权均方误差**：



这样就得到了IRLS：



为了找到最小的近似解，提出了IRLS：初始， 初始由（2）给出，用（1）式计算出新的误差,用来设置新的权值矩阵，然后将新的用在下一轮迭代中。这样我们找到了新的解，并重复，直到收敛。为了保证收敛，这个过程应该是一个收缩的图，最后收敛到一个固定点也就是解。MATLAB程序为：

% m-file IRLS0.m to find the optimal solution to Ax=b

% minimizing the L\_p norm ||Ax-b||\_p, using basic IRLS.

% csb 11/10/2012

function x = IRLS0(A,b,p,KK)

if nargin < 4, KK=10; end;

x = pinv(A)\*b; % Initial L\_2 solution

E = [];

for k = 1:KK % Iterate

e = A\*x - b; % Error vector

w = abs(e).^((p-2)/2); % Error weights for IRLS

W = diag(w/sum(w)); % Normalize weight matrix

WA = W\*A; % apply weights

x = (WA'\*WA)\(WA'\*W)\*b; % weighted L\_2 sol.

ee = norm(e,p); E = [E ee]; % Error at each iteration

end

plot(E)

这个核心想法一直在被改进。用这个基础形式，当时，算法能可靠地收敛。一个改进是部分地更新解：



其中是用（2）计算出的新的解，用来部分更新这一轮的。第二个改进是用特殊的更新因子



来显著提高收敛速率。当时，这就变成了Newton’s method的一个形式。但是一开始的收敛性会变得不鲁棒。应用了这些想法的MATLAB程序：

% m-file IRLS1.m to find the optimal solution to Ax=b

% minimizing the L\_p norm ||Ax-b||\_p, using IRLS.

% Newton iterative update of solution, x, for M > N.

% For 2<p<infty, use homotopy parameter K = 1.01 to 2

% For 0<p<2, use K = approx 0.7 - 0.9

% csb 10/20/2012

function x = IRLS1(A,b,p,K,KK)

if nargin < 5, KK=10; end;

if nargin < 4, K = 1.5; end;

if nargin < 3, p = 10; end;

pk = 2; % Initial homotopy value

x = pinv(A)\*b; % Initial L\_2 solution

E = [];

for k = 1:KK % Iterate

if p >= 2, pk = min([p, K\*pk]); % Homotopy change of p

else pk = max([p, K\*pk]); end

e = A\*x - b; % Error vector

w = abs(e).^((pk-2)/2); % Error weights for IRLS

W = diag(w/sum(w)); % Normalize weight matrix

WA = W\*A; % apply weights

x1 = (WA'\*WA)\(WA'\*W)\*b; % weighted L\_2 sol.

q = 1/(pk-1); % Newton's parameter

if p > 2, x = q\*x1 + (1-q)\*x; nn=p; % partial update for p>2

else x = x1; nn=2; end % no partial update for p<2

ee = norm(e,nn); E = [E ee]; % Error at each iteration

end

plot(E)

**5.3 未知数大于方程的欠定系统**

和5.2一样，不同在于：



MATLAB程序为：

% m-file IRLS2.m to find the optimal solution to Ax=b

% minimizing the L\_p norm ||x||\_p, using IRLS.

% Newton iterative update of solution, x, for M < N.

% For 2<p<infty, use homotopy parameter K = 1.01 to 2

% For 0<p<2, use K = approx 0.7 to 0.9

% csb 10/20/2012

function x = IRLS2(A,b,p,K,KK)

if nargin < 5, KK= 10; end;

if nargin < 4, K = .8; end;

if nargin < 3, p = 1.1; end;

pk = 2; % Initial homotopy value

x = pinv(A)\*b; % Initial L\_2 solution

E = [];

for k = 1:KK

if p >= 2, pk = min([p, K\*pk]); % Homotopy update of p

else pk = max([p, K\*pk]); end

W = diag(abs(x).^((2-pk)/2)+0.00001); % norm weights for IRLS

AW = A\*W; % applying new weights

x1 = W\*AW'\*((AW\*AW')\b); % Weighted L\_2 solution

q = 1/(pk-1); % Newton's parameter

if p >= 2, x = q\*x1 + (1-q)\*x; nn=p; % Newton's partial update for p>2

else x = x1; nn=1; end % no Newton's partial update for p<2

ee = norm(x,nn); E = [E ee]; % norm at each iteration

end;

plot(E)

**其它：**

**1.协方差矩阵：**

**方差：**一个变量的离散程度

**协方差：**两个变量的相关性

**协方差矩阵**：对维随机变量，共有个协方差，组成一个协方差矩阵。

**2.方向导数：方向导数是标量**。对曲面上某一点，可沿各个方向作切线。切线的斜率就是对应各个方向的方向导数。而对某一点的所有切线都在同一个平面上（称为切平面），因此某一点有且只有一个最陡峭的方向。由此引出梯度的定义：**梯度是一个矢量**，在其方向上方向导数最大，它的值就是这个方向导数。

**偏导数和方向导数的关系**：对多元函数，在某一点函数沿着某个坐标轴方向的导数就是偏导数，因此偏导数属于方向导数。

**3.矩阵特征值：**

1. 特征值和等于对角线元素和；
2. 特征值乘积等于矩阵的行列式；
3. 矩阵可逆等价于有n个非零特征值。
4. 特征值不同，则所对应的特征向量线性无关。

**特征向量的意义：**

矩阵左乘一个向量，是对这个向量进行伸缩或旋转变换，得到一个新向量。如果对某个向量只产生伸缩变换，则该向量称为矩阵的一个特征向量。式中对应于伸缩变换的程度，对应于这个特征向量。