基于混合模重构的 kNN 回归

龚永红[1,2] 宗 鸣[1,3] 朱永华[4] 程德波[1,3]

[1](广西师范大学广西多源信息挖掘与安全重点实验室 广西 桂林 541004)

[2](桂林航天工业学院图书馆 广西 桂林 541004)

[3](广西师范大学计算机科学与信息工程学院 广西 桂林 541004)

[4](广西大学计算机与电子信息学院 广西 南宁 530004)

摘 要 对于线性回归中 kNN(k-Nearest Neighbor)算法的 k 值固定问题和训练样本中的噪声问题,提出一种新的基于重构的稀疏编码方法。该方法用训练样本重构每一个测试样本,重构过程中,l_1-范数被用来确保每个测试样本被不同数目的训练样本来预测,以此解决 kNN 算法固定 k 值问题; $l_{2,1}$-范数导致的整行稀疏被用来去除噪声样本,以避免数据集上的噪声对重构产生不利影响。实验在 UCI 数据集上显示:新的改进算法比原来的 kNN 算法在线性回归中具有更好的预测效果。

关键词 线性回归 稀疏编码 重构 l_1-范数 $l_{2,1}$-范数 噪声样本

中图分类号 TP181 **文献标识码** A DOI: 10.3969/j.issn.1000-386x.2016.02.054

KNN REGRESSION BASED ON MIXED-NORM RECONSTRUCTION

Gong Yonghong[1,2] Zong Ming[1,3] Zhu Yonghua[4] Cheng Debo[1,3]

[1](*Guangxi Key Lab of Multi-source Information Mining and Security, Guangxi Normal University, Guilin 541004, Guangxi, China*)

[2](*Guilin University of Aerospace Technology Library, Guilin 541004, Guangxi, China*)

[3](*School of Computer Science and Information Technology, Guangxi Normal University, Guilin 541004, Guangxi, China*)

[4](*School of Computer, Electronics and Information, Guangxi University, Nanning 530004, Guangxi, China*)

Abstract This paper proposes a new reconstruction-based sparse coding method for solving the problem of k value fixing of k-NN in linear regression and the problem of noise in training samples. The method reconstructs every test sample using training sample. In reconstruction process, the l_1-norm is used to ensure each test sample will be predicted by training sample in different numbers and thus to solve the problem of k-NN algorithm in fixing k value, and the entire row sparse incurred by $l_{2,1}$-norm is used to remove noise samples so as to prevent the noise in dataset from adverse impact on the reconstruction. Experimental results on UCI datasets show that the new improved algorithm outperforms the previous k-NN regression method in terms of prediction effect.

Keywords Linear regression Sparse coding Reconstruction l_1-norm $l_{2,1}$-norm Noise sample

0 引 言

kNN 算法是一种应用广泛的分类方法,它是最近邻算法(NN 算法)的推广形式。NN 算法最早是由 Cover 和 Hart 在 1967 年提出,最早用于分类的研究[1]。kNN 算法也可用于回归:对于一个测试样本,在所有训练样本中选取最接近它的 k 个样本的均值来进行预测。然而本文发现传统的 kNN 算法对每一个测试样本,都用同样 k 个数目的训练样本来进行预测,这在应用中不合实际。

图 1 所示一个数据集分布,三角形代表测试样本,圆圈代表训练样本,如设 $k=3$,用 kNN 算法预测测试样本,对于左边的测试样本,用最邻近的三个训练样本去预测,这比较合理。但对于右边的测试样本,仅有两个训练样本离它比较近,另一个离得很远,显然用最邻近的这三个训练样本来预测测试样本是不合理的,应该根据实际情况,选取不同的 k 值。

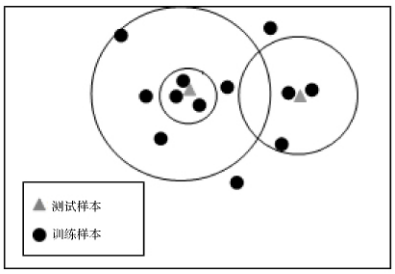

图 1 $k=3$ 时两个测试样本

收稿日期:2014-08-19。国家自然科学基金项目(61170131,61263035,61363009);国家高技术研究发展计划项目(2012AA011005);国家重点基础研究发展计划项目(2013CB329404);广西自然科学基金项目(2012GXNSFGA06004);广西研究生教育创新计划项目(YCSZ2015095,YCSZ2015096);广西多源信息挖掘与安全重点实验室开放基金项目(MIMS13-08)。龚永红,本科,主研领域:图书情报学,数据挖掘。宗鸣,硕士。朱永华,本科。程德波,硕士。

对此,一些学者展开了选取最优 k 值的研究。例如 Cora 等人提出了自动选取最优 k 值的 kNN 方法[2]。Matthieu Kowalski 在稀疏扩展方法里引入了结构化稀疏的概念[3],同时将这种方法与多层信号扩展方法联系起来,用来分解由许多不同成分构成的信号。Hechenbichler 等人提出了加权 kNN 法[4],该方法根据训练样本到测试样本距离的大小赋予不同的权值,距离大的权值反而小,该方法的分类识别率对 k 值的选取不再敏感。Zhang 等人提出了代价敏感分类方法[5-8]。还有一些学者从特征加权的角度提出了一些算法,也在一定程度上改进了 kNN 算法。

但是以上这些方法都是只利用了训练样本中 k 个邻近样本提供的信息,没有考虑测试样本提供的信息,即没有考虑训练样本和测试样本之间的相关性。而本文认为样本之间是存在相关性的,对于每个测试样本,应该用与之相关的训练样本来对其进行预测,但是每一个测试样本却可能跟不同数目的训练样本相关。为此,本文用训练样本重构[9]每一测试样本获得样本之间的 k 相关性,同时用 LASSO(the Least Absolute Shrinkage and Selection Operator) [10]来控制稀疏性以解决 k 值固定问题。另外实际数据集中一般会有噪声存在[11,12],如在图 1 中设 $k=7$,对于左边的测试样本,图中左上角的噪声样本会影响其真实值的预测,并应该剔除这些噪声样本,本文借助 $l_{2,1}$-范数能产生整行为 0 的特性来去除噪声。因此,针对 kNN 算法存在的这两个问题,本文提出一种新的基于混合模[13]重构的 kNN 算法—Mixed-Norm Reconstruction-kNN,简记为 MNR-kNN。

1　基于混合模重构的 kNN 算法

1.1　重　构

用训练样本重构每一测试样本时,本文假设有训练样本空间 $X\in R^{n\times d}$,n 为训练样本数目,d 为样本维数; 测试样本空间 $Y\in R^{d\times m}$,m 为测试样本数目。一般我们用最小二乘法[14,15]解决线性回归问题,即获取投影矩阵 $W\in R^{n\times m}$:

$$\min_W \sum_{i=1}^{m} \| y_i - X^{\mathrm{T}} w_i \|_2^2 = \min_W \| Y - X^{\mathrm{T}} W \|_F^2 \qquad (1)$$

其中,$\| \cdot \|_F$ 是 Frobenius 矩阵范数,$y_i \in R^{d\times 1}$,w_i 是 W 的第 i 列向量。

分类问题中 y_i 一般为类标签,W 表示 Y 与 X 的函数关系,在本文 y_i 表示第 i 个测试样本,而 W 表示训练样本和测试样本经过重构得到的相关性系数矩阵。以下面例子进一步说明 W,假设有 4 个训练样本,2 个测试样本,样本属性数目为 3,此时 $Y - X^{\mathrm{T}}W$ 为:

$$\begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \\ y_{31} & y_{32} \end{bmatrix} - \begin{bmatrix} x_{11} & x_{21} & x_{31} & x_{41} \\ x_{12} & x_{22} & x_{32} & x_{42} \\ x_{13} & x_{23} & x_{33} & x_{43} \end{bmatrix} \times \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \\ w_{41} & w_{42} \end{bmatrix} \qquad (2)$$

由 $X^{\mathrm{T}}W$ 知 W 中元素 W_{ij} 表示第 i 个训练样本与第 j 个测试样本之间的相关性大小,$W_{ij}>0$ 时代表正相关,$W_{ij}<0$ 时代表负相关,$W_{ij}=0$ 时代表不相关。所以,W 表示测试样本与训练样本之间的相关性矩阵,本文考虑训练样本和测试样本之间的相关性,利用重构方法获得了训练样本与测试样本之间的相关性大小。

1.2　正则化

一般得到的 W 不是稀疏的,考虑到列向量 W_j 表示第 j 个测试样本与所有训练样本的相关性,如果列向量中有 r 个元素值不为 0,其余为 0,则预测时 k 相应地取 r,并用对应的这 r 个训练样本来预测测试样本。这样选取的 k 个训练样本就是与测试样本最相关的 k 个样本,也就可以解决 kNN 算法中 k 值固定问题。

通常使用最小二乘损失函数求解线性回归问题,即:

$$\min_W \| X^{\mathrm{T}}W - Y \|_F^2 \qquad (3)$$

虽然上面目标函数是凸的,易知其解 $W^* = (XX^T)^{-1}XY$。然而,实际应用中 XX^{T} 不一定可逆,为此,优化函数式(3)被加上一正则化因子,即:

$$\min_W \| X^{\mathrm{T}}W - Y \|_F^2 + \rho \| W \|_2^2 \qquad (4)$$

其中,$\| W \|_2^2 = \sum_{i=1}^{n}\sum_{j=1}^{m} |w_{ij}|^2$,此时优化函数式(4)称为岭回归,其解为 $W^* = (XX^{\mathrm{T}} + \rho I)^{-1}XY$。但得到的 W 不稀疏,不能解决本文问题。因此本文用 l_1-范数和 $l_{2,1}$-范数取代式(4)中的 l_2-范数,得到以下目标函数:

$$\min_W \| X^{\mathrm{T}}W - Y \|_F^2 + \rho_1 \| W \|_1 + \rho_2 \| W \|_{2,1} \qquad (5)$$

其中,$\| W \|_1 = \sum_{i=1}^{n}\sum_{j=1}^{m} |w_{ij}|$,$\| W \|_{2,1} = \sum_{i=1}^{n} \left(\sum_{j=1}^{m} |w_{ij}|^2 \right)^{\frac{1}{2}}$,参数 ρ_1 调控 W 整体的稀疏性,ρ_2 调控 W 整行的稀疏性。l_1-范数已经被证明能使回归结果生成稀疏的回归稀疏,且这种稀疏是分布在矩阵中的元素,因此成为元组稀疏[11,14,15]。$l_{2,1}$-范数能导致回归优化出整行的稀疏,即行稀疏[16]。

通过 1.2 节的方法可以求解目标函数式(5)得到的 W,如下形式:

$$W = \begin{pmatrix} 0.2 & -0.5 & 0 \\ 0 & 0 & 0 \\ 0 & 0.8 & -0.9 \\ 0.11 & 0.25 & 0.3 \end{pmatrix}$$

其中,W 的第二行全为 0,即行稀疏。这是由于 $l_{2,1}$-范数导致的结果,这表明第二个训练样本跟所有测试样本无关。因此,第二个训练样本可能是噪音样本。此外第一列有两个非零值,即第一个和第四个,可以说第一个测试样本与第一、第四个训练样本相关。因此对第一个测试样本预测时 $k=2$。以此类推,如第二列有三个非零值,所以对第二个测试样本而言 $k=3$,对第四个测试样本 $k=2$。这就解决了 kNN 算法中 k 值固定问题,即对于不同测试样本 k 值是不一样的。而 kNN 算法中的 k 值通常由用户决定,本文每个测试样本的 k 值是通过稀疏学习得到的,是一种数据驱动分析方法。

1.3　MNR-kNN 算法

根据以上例子,目标函数式(5)利用 $l_{2,1}$-范数查找除了存在于训练集中的噪音样本,而且还利用 l_1-范数学习出与每个测试样本相关的训练样本。每个测试样本相关的训练样本的个数不同,即为 kNN 回归学习出了合适的 k。这样的学习方法是数据驱动的,也解决了本文提出 kNN 回归存在的两个问题,即噪音样本避免问题和固定 k 值问题。

本文预测测试样本时用 W 列的非零值对应的训练样本去预测,当然此时 k 的取值等于 W 相应列非零值的个数,这种方法本文称为不加权 MNR-kNN 算法。但是考虑到 W 中的元素值大小表示测试样本和训练样本的相关性大小,相应的训练样本

和测试样本之间的相关度大小是不同的，元素值越大，表明相关度越大；元素值越小，表明相关度越小。因此本文预测时根据 W_j 中的元素值对相应的训练样本做加权处理，可以得出第 j 个测试样本的加权预测值：

$$predictval\_weight = \sum_{i=1}^{n}\left(\frac{W_{ij}}{\sum_{i=1}^{n} W_{ij}} \times y_{train(i)}\right) \tag{6}$$

其中，$y_{train(i)}$ 表示第 i 个训练样本的真实值，即第 i 个训练样本的类标号。这种回归方法本文称为加权 MNR-kNN 算法。

最后，本文给出加权/不加权 MNR-kNN 算法的步骤，见算法 1。

算法 1 加权/不加权 MNR—kNN

输入：样本数据

输出：预测值

1) 将数据进行规范化处理

2) 通过算法 2(下面给出) 得到最优解 W

3) 加权/不加权情况下 MNR-kNN 算法对所有测试样本计算出相应的 k 值

4) 根据 k 值，通过式(6) 计算测试样本的预测值(注：不加权的情况相应的用 1 代替 W 中的非零值即可)

2 算法优化分析

虽然式(5) 是凸的，但后面两项正则化都是非光滑的。为此，本文提出一种有效的算法去求解目标函数。

具体地，首先对 $w_i(1 \leqslant i \leqslant m)$ 求导并命其为 0，可得：

$$XX^{\mathrm{T}}w_i - Xy^{(i)} + \rho_1 D_i w_i + \rho_2 \tilde{D} w_i = 0 \tag{7}$$

其中，$D_i(1 \leqslant i \leqslant m)$ 是对角矩阵，第 k 个对角元素为 $\frac{1}{2\|w_{ki}\|}$，$\tilde{D}_i$ 也是对角矩阵，第 k 个对角元素为 $\frac{1}{2\|w^k\|_2}$。所以：

$$w_i = (XX^{\mathrm{T}} + \rho_1 D_i + \rho_2 \tilde{D})^{-1} Xy^{(i)} \tag{8}$$

D_i 和 $\tilde{D}$ 依赖于 W，因此它们也是未知的。根据文献[16]，本文提出一种迭代算法去求解最优值 W，见算法 2。

算法 2：目标函数优化算法

输入：X，Y

输出：$W^{(t)} \in R^{n\times m}$

1) 初始化 $W^1 \in R^{n\times m}$，t = 1；

2)

do{

(1) 计算对角矩阵 $D_i^{(t)}(1 \leqslant i \leqslant m)$ 和 $\tilde{D}^{(t)}$，其中 $D_i^{(t)}$ 第 k 个对角元素为 $\frac{1}{2|w_{ki}^{(t)}|}$，$\tilde{D}^{(t)}$ 第 k 个对角元素为 $\frac{1}{2\|(w^{(t)})^k\|_2}$；

(2) For 每个 i $(1 \leqslant i \leqslant m)$，

$$w_i^{(t+1)} = (XX^{\mathrm{T}} + \rho_1 D_i^{(t)} + \rho_2 \tilde{D}^{(t)})^{-1} Xy^{(i)};$$

(3) $t = t + 1$; } until 收敛

定理 1 算法 2 在每次迭代中目标值减小。

证明 根据算法里的第 2 步可得到：

$$W^{(t+1)} = \min_W Tr(X^{\mathrm{T}}W - Y)^{\mathrm{T}}(X^{\mathrm{T}}W - Y) + \rho_1 \sum_{i=1}^{m} w_i^{\mathrm{T}} D_i^{(t)} w_i + \rho_2 Tr W^{\mathrm{T}} \tilde{D}^{(t)} W \tag{9}$$

因此有：

$$Tr(X^{\mathrm{T}}W^{(t+1)} - Y)^{\mathrm{T}}(X^{\mathrm{T}}W^{(t+1)} - Y) + \rho_1 \sum_{i=1}^{m} (w_i^{(t+1)})^{\mathrm{T}} D_i^{(t)} w_i^{(t+1)} + \rho_2 Tr(W^{(t+1)})^{\mathrm{T}} \tilde{D}^{\mathrm{T}} W^{(t+1)}$$

$$\leqslant Tr(X^{\mathrm{T}}W^{(t)} - Y)^{\mathrm{T}}(X^{\mathrm{T}}W^{(t)} - Y) + \rho_1 \sum_{i=1}^{m} (w_i^{(t)})^{\mathrm{T}} D_i^{(t)} w_i^{(t)} + \rho_2 Tr(W^{(t)})^{\mathrm{T}} \tilde{D}^{\mathrm{T}} W^{(t)}$$

$$\Rightarrow Tr(X^{\mathrm{T}}W^{(t+1)} - Y)^{\mathrm{T}}(X^{\mathrm{T}}W^{(t+1)} - Y) + \rho_1 \sum_{i=1}^{d}\sum_{j=1}^{m}\left(\frac{(w_{ij}^{(t+1)})^2}{2\|w_{ij}^{(t)}\|} - \|w_{ij}^{(t+1)}\| + \|w_{ij}^{(t+1)}\|\right) + \rho_2 \sum_{k=1}^{d}\left(\frac{\|(w^{(t+1)})^k\|_2^2}{2\|(w^{(t)})^k\|_2} - \|(w^{(t+1)})^k\|_2 + \|(w^{(t+1)})^k\|_2\right)$$

$$\leqslant Tr(X^{\mathrm{T}}W^{(t)} - Y)^{\mathrm{T}}(X^{\mathrm{T}}W^{(t)} - Y) + \rho_1 \sum_{i=1}^{d}\sum_{j=1}^{m}\left(\|w_{ij}^{(t)}\| + \frac{(w_{ij}^{(t)})^2}{2\|w_{ij}^{(t+1)}\|} - \|w_{ij}^{(t)}\|\right) + \rho_2 \sum_{k=1}^{d}\left(\|(w^{(t)})^k\|_2 + \frac{\|(w^{(t)})^k\|_2^2}{2\|(w^{(t)})^k\|_2} - \|(w^{(t)})^k\|_2\right)$$

$$\Rightarrow Tr(X^{\mathrm{T}}W^{(t+1)} - Y)^{\mathrm{T}}(X^{\mathrm{T}}W^{(t+1)} - Y) + \rho_1 \sum_{i=1}^{d}\sum_{j=1}^{m}\|w_{ij}^{(t+1)}\| + \rho_2 \sum_{k=1}^{d}\|(w^{(t+1)})^k\|_2 \leqslant Tr(X^{\mathrm{T}}W^{(t)} - Y)^{\mathrm{T}}(X^{\mathrm{T}}W^{(t)} - Y) + \rho_1 \sum_{i=1}^{d}\sum_{j=1}^{m}\|w_{ij}^{(t)}\| + \rho_2 \sum_{k=1}^{d}\|(w^{(t)})^k\|_2$$

根据文献[17]对于任意向量 w 和 w_0，有 $\|w_2\| - \frac{\|w\|_2^2}{2\|w_0\|_2} \leqslant \|w_0\|_2 - \frac{\|w_0\|_2^2}{2\|w_0\|_2}$。因此最后一步成立，即算法在每次迭代中减小目标值。

$W^{(t)}$、$D_i^{(t)}(1 \leqslant i \leqslant m)$ 和 $\tilde{D}^{(t)}$ 在收敛处满足式(9)。由于式(7) 是一个凸问题，满足式(9) 意味着 W 对于式(7) 来说是一个全局最优解。因此算法 2 将收敛到式(7) 的全局最优解。因为在每一次迭代时有封闭形式解，所以本文提出的算法收敛非常快。

3 实验结果与分析

实验数据来自 UCI 机器学习库[18]，具体细节见表 1。

表 1 数据集基本情况

数据集	实例数	属性数
ConcreteSlump(CS)	103	10
Housing(HS)	506	14
Winequality(wine)	1599	12
Mpg	398	8

本次实验主要是比较 kNN 算法、不加权 MNR-kNN 算法和加权 MNR-kNN 算法这三种算法的预测效果，本文选用经典评价指标 RMSE 和相关系数 CorrCoef。RMSE 和 CorrCoef 定义分别如下：

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{10}$$

$$CorrCoef = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}}_i)(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}}_i)^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (11)$$

其中，n 为样本个数，y_i 为真实值，$\hat{y}_i$ 为预测值。

RMSE 一般用来作为算法效果的评判依据，通常 RMSE 越小，预测值和真实值之间的偏差就越小，算法的效果也就越好，否则越差。CorrCoef 表示预测值和真实值之间的相关性大小，一般相关性越大，预测越准确，反之相反。另外本文用 Matlab 编程实现具体程序代码，在每个数据集上用 10 折交叉验证法做十次实验来看算法效果。为了保证公平性，kNN 算法、不加权 MNR-kNN 算法和加权 MNR-kNN 算法在每一次实验中选用相同的训练集和测试集，记录这三种算法十次实验取得的 RMSE 和 CorrCoef 情况。

图 2—图 5 为四个数据集上的实验结果，纵坐标为 RMSE 的大小，横坐标为十次实验次序。

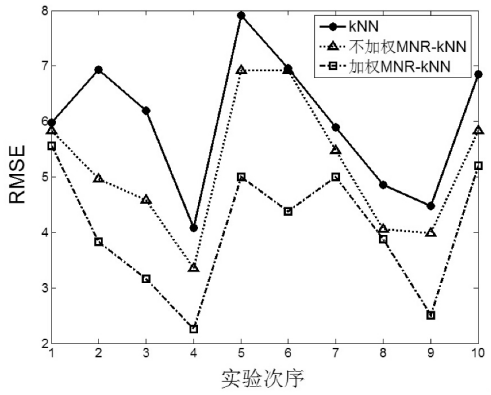

图 2　数据集 ConcreteSlump

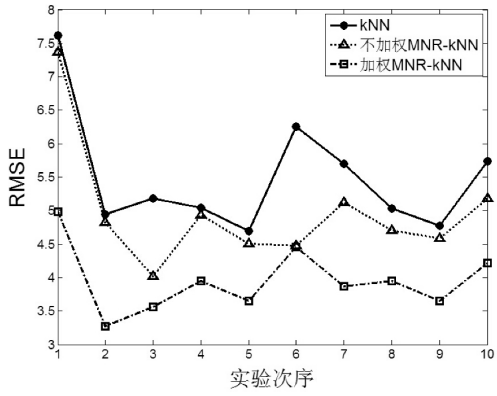

图 3　数据集 Housing

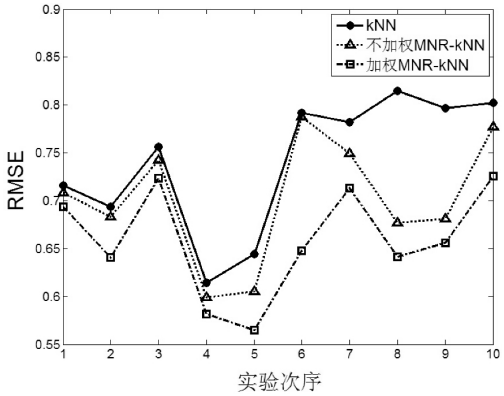

图 4　数据集 Winequailty

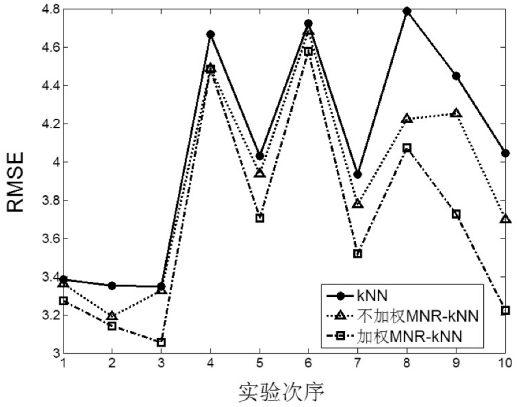

图 5　数据集 Mpg

图 6—图 9 为四个数据集上的实验结果，纵坐标为 CorrCoef 的大小，横坐标为十次实验次序。

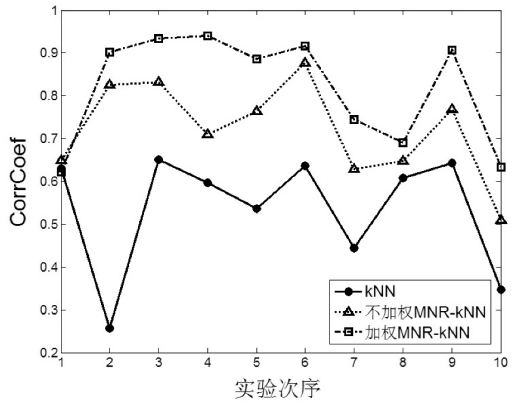

图 6　数据集 ConcreteSlump

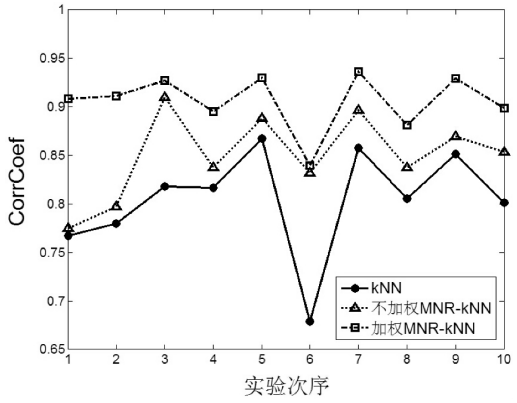

图 7　数据集 Housing

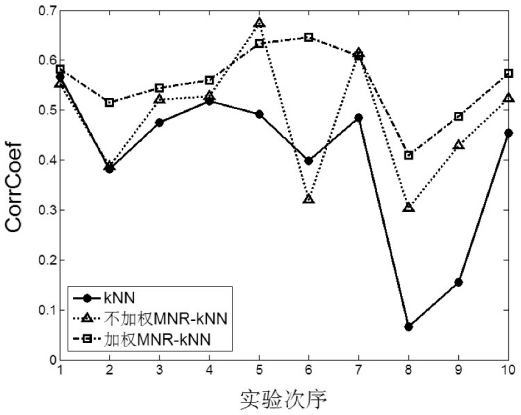

图 8　数据集 Winequailty

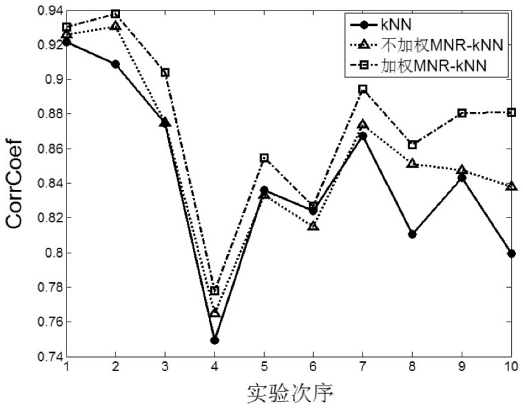

图9 数据集 Mpg

RMSE 和 CorrCoef 的均值以及方差见表 2 和表 3。

表 2 RMSE 均值和方差情况

Dataset	kNN	不加权 MNR-kNN	加权 MNR-kNN
CS	6.0126 ±1.3476	5.1913 ±1.3377	4.0766 ±1.1965
HS	5.4979 ±0.7176	4.9710 ±0.7455	3.9559 ±0.2176
Wine	0.7413 ±0.0045	0.7011 ±0.0038	0.6590 ±0.0028
Mpg	4.0742 ±0.2982	3.8958 ±0.2371	3.6794 ±0.2704

表 3 CorrCoef 均值和方差情况

Dataset	kNN	不加权 MNR-kNN	加权 MNR-kNN
CS	0.5350 ±0.0175	0.7212 ±0.0116	0.8176 ±0.0152
HS	0.8043 ±0.0027	0.8498 ±0.0017	0.9058 ±0.0008
Wine	0.3392 ±0.0238	0.4858 ±0.0135	0.5565 ±0.0046
Mpg	0.8434 ±0.0024	0.8555 ±0.0022	0.8750 ±0.0021

图 2 – 图 9 在四个 UCI 数据集上的实验显示，对于评价指标 RMSE，加权 MNR-kNN 算法得出的 RMSE 均值最小，其次是不加权 MNR-kNN 算法，RMSE 最大的是 kNN 算法。对于评价指标 CorrCoef，总体看来，加权 MNR-kNN 算法得出的 CorrCoef 均值最大，其次是不加权 MNR-kNN 算法，CorrCoef 最大的是 kNN 算法。

根据表 2，对于评价指标 RMSE，加权 MNR-kNN 算法取得的 RMSE 均值最小，其次分别是不加权 MNR-kNN 算法和 kNN 算法。最明显的在 CS 数据集上，加权 MNR-kNN 算法比 kNN 算法在评价指标 RMSE 均值上降低了 1.936，同时不加权 MNR-kNN 算法也比 kNN 算法降低了 0.8213。另外根据表 3，加权 MNR-kNN 算法取得的相关系数均值最大，其次是不加权 MMR-kNN 算法和 kNN 算法。在 CS 数据集上的实验表明，加权 MNR-kNN 算法和不加权 MNR-kNN 算法比 kNN 算法在评价指标上 CorrCoef 均值上分别提高了 28.26% 和 18.62%，改善效果显著。此外，从 RMSE 和 CorrCoef 方差总体情况来看，加权 MNR-kNN 算法方差最小，其次是不加权 MMR-kNN 算法和 kNN 算法。这说明本文提出的 MNR-kNN 算法比 kNN 算法算法更稳定。

因此，综合看来，加权 MNR-kNN 算法效果最好，其次是不加权 MMR-kNN 算法，效果最差的是 kNN 算法。这说明本文提出的 MMR-kNN 算法（包括加权和不加权两种情况）预测准确率更高，效果相比 kNN 算法也更好，同时加权 MNR-kNN 算法比不加权 MNR-kNN 算法要好。

4 结 语

针对 kNN 算法中 k 值固定不变问题以及如何在预测时去除噪声样本，本文提出了基于混合模重构的 kNN 算法（MNR-kNN）。通过 l_1-范数，MNR-kNN 算法中 k 值根据样本之间相关性情况取不同的值，同时在重构过程中利用 $l_{2,1}$-范数去除噪声。另外根据相关性大小可将 MNR-kNN 算法具体分加权情况和不加权情况。实验表明，在同样的训练样本和测试样本情况下，使用三种算法进行预测，加权 MNR-kNN 算法效果最好，其次是不加权 MNR-kNN 算法，而 kNN 算法取得的效果不佳。

本文提出的 MNR-kNN 算法可以用于信号编码和压缩、降维去噪、人脸识别、文本挖掘等领域。将来可以进一步考虑如何在实际应用中降低 MNR-kNN 算法的时间复杂度，比如分块处理等，以此来取得相应的实际应用价值。

参 考 文 献

[1] Qin Y S, Zhang S C, Zhang C Q. Combinining knn imputation and bootstrap calibreated: empirical likelihood for incomplete data analysis [J]. International Journal of Data Warehousing and Mining, 2010, 6(4): 61-73.

[2] Cora G, Wojna A. A classifier combining rule induction and KNN method with automated selection of optimal neighbourhood [C]//Proceedings of the 13rd European Conference on Machine Learning. London, UK: Springer-Verlag, 2002: 111-123.

[3] Matthieu Kowalski. Sparse regression using mixed norms [J]. Applied and Computational Harmonic Analysis, 2009, 27(3): 303-324.

[4] Hechenbichler K, Schliep K. Weighted k-nearest-neighbor techniques and ordinal classification, Discussion Paper 399 [R]. Munich, Germany: Ludwing-Maximilians University Munich, 2004.

[5] Wang T, Qin Z X, Zhang S C, et al. Cost-sensitive classification with inadequate labeled data [J]. Information Systems, 2012, 37(5): 508-516.

[6] Zhang S C. Decision tree classifiers sensitive to heterogeneous costs [J]. Journal of Systems and Software, 2012, 85(4): 771-779.

[7] Zhang S C. Cost-sensitive classification with respect to waiting cost [J]. Knowledge-Based Systems, 2010, 23(5): 369-378.

[8] Zhu X, Zhang S, Zhang J, et al. Cost – sensitive imputing missing values ordering [J]. Aaai Press, 2007, 2: 1922-1923.

[9] 李长彬，黄东，黎永壹. 局部线性重构分类算法研究 [J]. 计算机应用与软件，2013，30(4)：156-158，210.

[10] Zhu X F, Huang Z, Cheng H, et al. Sparse hashing for fast multi-media search [J]. ACM Transactions on Information System, 2013, 31(2): 9.

[11] Zhang S C, Jin Z, Zhu X F. Missing data imputation by utilizing information within incomplete instances [J]. Journal of System and Software, 2011, 84(3): 452-459.

[12] Zhu X F, Zhang S C, Jin Z, et al. Missing value estimation for mixed-attribute data sets [J]. IEEE Transactions on Knowledge and Engineering, 2011, 23(1): 110-121.

[13] Zhu X F, Huang Z, Shen H T, et al. Dimensionality reduction by Mixed Kernel Canonical Correlation Analysis [J]. Pattern Recognition, 2012, 45(8): 3003-3016.

[14] Zhu X F, Suk H, Shen D. A Novel Matrix – Similarity Based Loss Function for Joint Regression and Classification in AD Diagnosis [J]. NeuroImage, 2014, 100: 91-105.

（下转第 241 页）

其中 1 代表等效点权值算法,2 代表节点综合测度算法,3 代表基于相似度贡献的节点重要性评估算法。

可以看出: 各个算法得到结果的相似度都很高,基本上都在 80% 以上,其中算法 1 和算法 3 的评价结果是最相似的,它们分别是从节点删除后造成的损失和节点相似性两个完全不同的方面来研究的,但是评价出的结果却基本一致。所以基于 F-度量方法的结果,最后选取算法 1 和算法 3 共同的排序结果,其中前五名如表 6 所示。

表 6 民航旅客社会网络的前五名重要节点

名次	节点编号	节点的 BOOK_ID
1	1	28708788
2	20	28837334
3	11	33930659
4	14	29407420
5	21	29405672

3.3 结果分析

实验分别从节点度、点权、二重度、二重点权以及实际数据库中节点 1 月份出行情况两个方面,对排序结果进行了可靠性分析,排序结果见表 7 所示。

表 7 节点度,点权,二重度,二重点权的排序结果

名次	节点编号	度	点权	二重度	二重点权
1	1	38	116	276	887
2	20	38	116	276	887
3	11	18	59	243	765
4	14	18	59	243	765
5	21	18	59	243	765

其中,度是指与该节点相邻的所有节点的个数; 点权是指与该节点相邻的所有节点的权重之和; 二重度是指与节点相邻的所有节点的度之和; 二重点权是指与节点相邻的所有节点的点权之和。

从表 7 可以看出: 分别按节点度、点权、二重度、二重点权排序的结果的前五名也正好是 3.2 节中所求的五个重要节点。因此,进一步地说明了最后所求出的重要节点是准确的。

4 结 语

本文主要从民航旅客社会网络的构建以及重要节点发现算法在民航旅客社会网络的应用方面展开研究,通过挖掘 PNR 数据间的内在关系,进一步推测出民航旅客间的关系以及这种关系的强弱,从而构建出民航旅客社会网络。在此基础上分析了多种社会网络重要节点发现算法,并进一步地把这些算法应用在了民航旅客社会网络中,最后通过 F-度量方法对结果进行了相似性比较,确定出重要的民航旅客。实验结果证明,本次研究为民航旅客社会网络分析与挖掘提供了社会网络类型的数据支持,并更加全面准确地分析了民航旅客,为提高航空公司效益,改善民航旅客服务提供了有力依据。

参 考 文 献

[1] 王红,李晓辉. 基于数据挖掘的航空公司客户信息分析[J]. 计算机工程,2005,31(S1):189-191.
[2] 冯霞,李勇,陈卉敏. 民航旅客社会网络构建方法研究[J]. 计算机仿真,2013,30(6):51-54.
[3] 冯霞,陈卉敏,李勇. 一种结合 SVD 的 SNMF 复杂网络社区发现方法[J]. 信息与控制,2013,42(3):387-391.
[4] 潘玲玲. 基于旅客行为的航空旅客细分模型研究及其实现[D]. 南京: 南京航空航天大学,2012.
[5] 罗利,彭际华. 竞争环境下的民航客运收益管理动态定价模型[J]. 系统工程理论与实践,2007(11):15-24.
[6] Barabasi A L,Albert R. Emergence of scaling in random networks[J]. Science,1999,286(5439):509-512.
[7] Watts D J,Strogatz S H. Collective dynamics of small world networks[J]. Nature,1998,393(4):440-442.
[8] Barabasi A L,Bonabeau E. Scale-Free networks[J]. Scientific American,2003,288(5):60-69.
[9] 杨育彬,李宁,张瑶. 基于社会网络可视化分析的数据挖掘[J]. 软件学报,2008,19(8):1980-1994.
[10] Tang J T,Wang T,Wang J. Shortest Path Approximate Algorithm for Complex Network Analysis[J]. Journal of software,2011,22(10):2279-2290.
[11] 乔少杰,唐常杰,彭京,等. 基于个性特征仿真邮件分析系统挖掘犯罪网络核心[J]. 计算机学报,2008,31(10):1795-1803.
[12] 王昊翔,曾珊,刘挥扬. 虚拟社交网络中节点重要度分析[J]. 上海交通大学学报,2013,47(7):1055-1059.
[13] 王喆. 基于复杂网络的社区发现算法研究[D]. 吉林: 吉林大学,2013.
[14] 司晓静. 复杂网络中节点重要性排序的研究[D]. 西安: 西安电子科技大学,2012.
[15] Opsahl T,Agneessens F,Skvoretz J. Node centrality in weighted networks: Generalizing degree and shortest paths[J]. Social Networks,2010,32:245-251.
[16] Wagner C,Roessner J,Bobb K,et al. Approaches to understanding and measuring interdisciplinary scientific research(IDR): A review of the literature[J]. Journal of Informatics,2011,5(1):14-26.
[17] Okamoto K,Chen W,Li X. Ranking of closeness centrality for large-scale social networks[J]. Lecture Notes in Computer Science,2008,5059:186-195.
[18] Kim H,Tang J,Aderson R,et al. Centrality prediction in dynamic human contact networks[J]. Computer Networks,2012,56(3):983-996.

(上接第 236 页)

[15] Zhu X F,Huang Z,Shen H T,et al. Linear Cross-Modal Hashing for Effective Multimedia Search[C]//Proceedings of ACM M M,2013:143-152.
[16] Zhu X F,Huang Z,Yang Y,et al. Self-tau-ght dimensionality reduction on the high-dimensional small-sized data[J]. Pattern Recognition,2013,46(1):215-229.
[17] Zhu X F,Wu X D,Ding W,et al. Feature selection by joint graph sparse coding[C]// Proceedings of the 2013 Siam Intern-ational Conference on Data Mining,2013:803-811.
[18] UCI repository of machine learning datasets[DB/OL]. [2012-01-15]. http://archive.ics.uci.edu/-ml/.