文章编号: 1000-5862(2012)06-0622-05

改进的偏最小二乘回归推荐算法

廖春华, 杜建强, 程春雷, 李智彪

(江西中医学院计算机学院, 江西 南昌 330004)

摘要: 基于已有的相关 PLS 算法, 提出了针对 QSAR 研究和工业过程控制建模的环境要求的 PLS 回归改进算法: 加强递归 PLS 算法. 模拟实验结果表明: 在实时建模过程中, 该算法的性能优于传统的 PLS 回归算法.

关键词: 偏最小二乘法回归; kernel 算法; 算法改进; 加权递归算法

中图分类号: O 625.63　　**文献标志码**: A

0 引言

在科研实践中, 经常遇到需要研究 2 组多重相关变量之间的相互依赖关系, 并研究用一组变量(常称为自变量、预测变量或者解释变量)去预测另一组变量(常称为因变量、响应变量或者反应变量). 当 2 组变量的数量很多时, 为了完备地描述系统, 尽可能多地保留一些重要的系统特征数据, 研究人员往往倾向于尽量多地选取有关指标, 这样构造的多变量系统必然会出现变量的多重相关性. 王惠文等[1]对于多重线性相关在回归建模中的危害作用进行了总结. 特别地, 由于受到各种客观因素(如实验周期、成本等)的影响, 当观测数据的样本数较小时, 基于最小二乘回归的方法(如多元线性回归(MLR)、主成分分析(PCA)和典型相关分析(CCA)等)都无法消除多重共线性带来的影响. 但是, 可以借鉴 PCA 和 CCA 的相关思想, 结合 PCA 的成分提取和 CCA 的解释变量与反应变量之间的相关性理论, 这就是偏最小二乘回归方法(PLSR).

PLS 方法首先由经济计量学家 Herman Wold 于 1966 年提出并用于社会科学领域的研究[2]. 1983 年, S.Wold 等[3]提出了基于 PLS 的一种新型多元统计分析方法——偏最小二乘回归(partial least-squares regression, PLSR). 此后 PLSR 在计量化学、生物、医学等领域得到广泛应用和发展. Wold[4]、H. Skuldson[5]、P. Geladi[6]和 M. Tenenhaus[7] 等研究指出, 在反应变量对多解释变量的回归建模中, 当各变量集合部存在较高程度的相关性时, 用偏最小二乘回归建模分析, 比一般多元回归分析更加有效, 其结论更加可靠. 王惠文[8]总结了 PLSR 的主要特点: 一方面通过数据分析寻找反应变量与解释变量之间的函数关系, 建立模型进行预测; 另一方面, 通过数据分析简化数据结构, 观察变量间的相互关系. 因此, C. Fornell[9]将偏最小二乘回归方法称为第 2 代回归分析方法.

1 基本原理与算法研究

为了说明问题方便, 将本文所用的符号及含义列出如下: $\boldsymbol{X}$: 预测变量矩阵($n\times p$), $\boldsymbol{Y}$: 响应变量矩阵($n\times q$), $\boldsymbol{B}_{\mathrm{PLS}}$: PLS 回归系数矩阵($p\times q$), $\boldsymbol{W}$: $\boldsymbol{X}$ 矩阵的 PLS 的权重矩阵($p\times m$), $\boldsymbol{P}$: $\boldsymbol{X}$ 矩阵的 PLS 荷载矩阵($p\times m$), $\boldsymbol{Q}$: $\boldsymbol{Y}$ 矩阵的 PLS 荷载矩阵($q\times m$), $\boldsymbol{R}$: 直接从原始矩阵 $\boldsymbol{X}$ 计算得分 $\boldsymbol{T}$ 的 PLS 的权重矩阵($p\times m$), $\boldsymbol{T}$: PLS 得分数矩阵($n\times m$), $\boldsymbol{w}_a$: $\boldsymbol{W}$ 的列向量, $\boldsymbol{p}_a$: $\boldsymbol{P}$ 的列向量, $\boldsymbol{q}_a$: $\boldsymbol{Q}$ 的列向量, $\boldsymbol{r}_a$: $\boldsymbol{R}$ 的列向量, $\boldsymbol{t}_a$: $\boldsymbol{T}$ 的列向量, p: $\boldsymbol{X}$ 矩阵的变量数(列数), q : $\boldsymbol{Y}$ 矩阵的变量数(列数), n: 对象数(样本数), m: PLS 模型中成分的数量, h: 潜变量维数的整数计数器.

1.1 PLS 回归原理

PLS 回归要求在解释变量空间里寻找某种线性组合, 以能更好地解释反应变量的变异信息. 但 PLS 回归不是直接建立这种线性回归模型, 而是建

收稿日期: 2012-09-11

基金项目: 国家“973”计划(2010CB530602, 2010CB530603), 国家“863”计划 (2012AA02A609)和国家自然科学基金(81160424)资助项目.

作者简介: 廖春华(1972-), 江西南昌人, 讲师, 硕士, 主要从事形式化方法的研究.

立解释潜变量关于反应潜变量的线性回归模型, 间接反映解释变量与反应变量之间的关系, 即同时从解释变量与反应变量中提取 2 组潜变量, 它们分别是解释变量与反应变量的线性组合, 称为因子或成分. 这种提取要求满足 2 个要求: ①2 组潜变量分别最大限度地表示了各自的变量信息; ②对应的解释潜变量与反应潜变量之间的协方差最大化, 即 2 个潜变量的相关程度最大, 其数学模型可以表述为

$$\max < X_1 w_1, Y_1 c_1 >= w_1' X_1' Y_1 c_1,$$
$$s.t. \begin{cases} w_1' w_1 = 1, \\ c_1' c_1 = 1. \end{cases}$$

为了消除数据量纲引起的数据之间的波动, 假设所得到的解释变量矩阵 $\boldsymbol{X}$ 和反应变量矩阵 $\boldsymbol{Y}$ 都进行了中心化和标准化处理, 可以利用经典的 NIPALS 算法或者 kernel 算法[10]来得到一个 PLS 回归模型, A. Hoskuldsson[11]讨论了其中 PLS 潜变量的向量计算方法. 其方法是在连续的过程中, 每次分别计算出 $\boldsymbol{X}$ 矩阵和 $\boldsymbol{Y}$ 矩阵各一个潜在向量, 然后在下一次计算中以退化的 $\boldsymbol{X}$ 矩阵和 $\boldsymbol{Y}$ 矩阵代替原来的 $\boldsymbol{X}$ 矩阵和 $\boldsymbol{Y}$ 矩阵计算新的潜在向量.

以下是典型的 PLS 回归算法描述:

①$\boldsymbol{X}$ 矩阵和 $\boldsymbol{Y}$ 矩阵标准化.

②应用 NIPALS 算法或 kernel 算法计算 $\boldsymbol{w}$、$\boldsymbol{t}$、$\boldsymbol{q}$、$\boldsymbol{u}$ 和 $\boldsymbol{p}$.

③分别从 $\boldsymbol{X}$ 矩阵和 $\boldsymbol{Y}$ 矩阵中减去潜变量, 得到退化的 $\boldsymbol{X}$ 矩阵和 $\boldsymbol{Y}$ 矩阵

$$\boldsymbol{X}_{h+1} = \boldsymbol{X}_h - \boldsymbol{t}_h \boldsymbol{p}_h^{\mathrm{T}}, \tag{1}$$

$$\boldsymbol{Y}_{h+1} = \boldsymbol{Y}_h - \boldsymbol{t}_h \boldsymbol{q}_h^{\mathrm{T}}. \tag{2}$$

④计算预测残差平方和 PRESS(Predicted Residual Sum of Squares), 采用以上步骤, 分别计算去掉第 l 个样本点后用余下的 $n-1$ 个观测值按 PLS 回归方式建模, 并考虑在抽取 h 个成分后拟合的回归式, 然后把舍去的第 l 个观测点代入拟合的回归式, 得到 $\boldsymbol{y}_j$ 在第 l 个观测点上的预测值 $\hat{\boldsymbol{y}}_{\boldsymbol{j}(\boldsymbol{h})(-\boldsymbol{l})}$, 对 l 重复以上的验证, 即得到抽取 h 个成分时反应变量 $\boldsymbol{Y}$ 的预测误差平方和

$$PRESS_{(h)} = \sum_{j=1}^{q} \sum_{i=1}^{n} (\boldsymbol{Y}_{ij} - \hat{\boldsymbol{y}}_{j(h)(-l)})^2.$$

⑤如果 $PRESS_{(h)} - PRESS_{(h-1)} < \varepsilon, \varepsilon$ 为预测精度, 则转入第⑥步; 否则转入第②计算下一个潜变量.

⑥建立 $\boldsymbol{Y}$ 关于 $\boldsymbol{r}$ 的线性回归方程

$$\boldsymbol{Y} = \boldsymbol{t}_1 \boldsymbol{r}_1' + \boldsymbol{t}_2 \boldsymbol{r}_2' + \cdots + \boldsymbol{t}_h \boldsymbol{r}_h' + \boldsymbol{F}.$$

⑦建立 PLS 回归方程. 由于 $\boldsymbol{t}_h$ 是 $\boldsymbol{X}$ 的线性组合, 因此可以通过逆标准化变换得到 PLS 回归方程

$$\boldsymbol{y}_j^* = \boldsymbol{a}_{jp} \boldsymbol{x}_1^* + \cdots + \boldsymbol{a}_{jp} \boldsymbol{x}_p^* + \boldsymbol{F}_{Mj}, \quad \boldsymbol{j} = 1, 2, \cdots, q,$$

$\boldsymbol{F}_{Mj}$ 为残差矩阵 $\boldsymbol{F}_M$ 的第 j 列.

通常, 在 PLS 回归过程中, $\boldsymbol{X}$ 矩阵和 $\boldsymbol{Y}$ 矩阵在计算每一个潜向量之后在步骤③进行退化.

1.2　PLS 回归的 kernel 算法的发展

方程(1)和方程(2)是一个关于解释矩阵和反应矩阵不断退化的过程. 因此, 当解释向量和反应向量的变量数量特别大时, 应该设法避免在计算过程中的 $\boldsymbol{X}$ 矩阵和 $\boldsymbol{Y}$ 矩阵的退化运算. 尤其当潜变量数较大时, 这种计算会经过多次迭代.

S.D. Bhupinder[12]证明只需要对解释变量矩阵 $\boldsymbol{X}$ 和反应变量矩阵 $\boldsymbol{Y}$ 这两者中的一个进行退化就可以满足算法的需要了. 也即

$$\boldsymbol{X}_{h+1}^{\mathrm{T}} \boldsymbol{Y}_{h+1} = \boldsymbol{X}_h^{\mathrm{T}} \boldsymbol{Y}_{h+1} = \boldsymbol{X}_{h+1}^{\mathrm{T}} \boldsymbol{Y}_h,$$
$$\boldsymbol{X}_{h+1}^{\mathrm{T}} \boldsymbol{X}_{h+1} = \boldsymbol{X}_{h+1}^{\mathrm{T}} \boldsymbol{X}_h.$$

PLS 回归的 kernel 算法最早由 F. Lindgren 等[10]提出. 假设已对解释矩阵 $\boldsymbol{X}$ 和反应矩阵 $\boldsymbol{Y}$ 作了以 0 为均值和 1 为方差的处理. 在 PLS 回归中, 可以证明 $\boldsymbol{X}_{h+1}^{\mathrm{T}} \boldsymbol{Y}_{h+1}$ 和 $\boldsymbol{X}_{h+1}^{\mathrm{T}} \boldsymbol{X}_{h+1}$ 不需要进行退化运算. 矩阵 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 的退化由方程(1)和方程(2)给出. 由此, $\boldsymbol{X}_{h+1}^{\mathrm{T}} \boldsymbol{Y}_{h+1}$ 和 $\boldsymbol{X}_{h+1}^{\mathrm{T}} \boldsymbol{X}_{h+1}$ 可以由

$$\boldsymbol{X}_{h+1}^{\mathrm{T}} \boldsymbol{Y}_{h+1} = (\boldsymbol{I} - \boldsymbol{w}_h \boldsymbol{p}_h^{\mathrm{T}})^{\mathrm{T}} \boldsymbol{X}_h^{\mathrm{T}} \boldsymbol{Y}_h, \tag{3}$$

$$\boldsymbol{X}_{h+1}^{\mathrm{T}} \boldsymbol{X}_{h+1} = (\boldsymbol{I} - \boldsymbol{w}_h \boldsymbol{p}_h^{\mathrm{T}})^{\mathrm{T}} \boldsymbol{X}_h^{\mathrm{T}} \boldsymbol{X}_h (\boldsymbol{I} - \boldsymbol{w}_h \boldsymbol{p}_h^{\mathrm{T}}) \tag{4}$$

计算, 由方程(3)和方程(4)可知, 在提取特征向量的计算中, $\boldsymbol{X}^{\mathrm{T}} \boldsymbol{X}$ 和 $\boldsymbol{X}^{\mathrm{T}} \boldsymbol{Y}$ 可以不用进行退化, 只需要进行迭代乘法运算. 方程(3)和方程(4)的退化计算中包含了矩阵的乘法运算, 计算量相对较大, 为此, De Jong 等[13]针对上述算法做了如下改进

$$(\boldsymbol{X}^{\mathrm{T}} \boldsymbol{X})_{h+1} = (\boldsymbol{X}^{\mathrm{T}} \boldsymbol{X})_h - \boldsymbol{p}_h \boldsymbol{p}_h^{\mathrm{T}} (\boldsymbol{t}_h^{\mathrm{T}} \boldsymbol{t}_h), \tag{5}$$

$$(\boldsymbol{X}^{\mathrm{T}} \boldsymbol{Y})_{h+1} = (\boldsymbol{X}^{\mathrm{T}} \boldsymbol{Y})_h - \boldsymbol{p}_h \boldsymbol{q}_h^{\mathrm{T}} (\boldsymbol{t}_h^{\mathrm{T}} \boldsymbol{t}_h). \tag{6}$$

由此可知, 退化阶段的矩阵乘法运算可以通过迭代的向量乘积来完成, 从而减少了计算工作量. 显然这种改进了的 kernel 算法比原始的 kernel 算法更快.

以下给出 kernel 的算法过程:

①计算协方差矩阵 $\boldsymbol{X}^{\mathrm{T}} \boldsymbol{X}$ 和 $\boldsymbol{X}^{\mathrm{T}} \boldsymbol{Y}$. kernel 矩阵 $\boldsymbol{X}^{\mathrm{T}} \boldsymbol{Y} \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{X}$ 可以由 $\boldsymbol{X}^{\mathrm{T}} \boldsymbol{Y}$ 乘以 $(\boldsymbol{X}^{\mathrm{T}} \boldsymbol{Y})^{\mathrm{T}}$ 计算得到.

②PLS 的权重向量 $\boldsymbol{w}_h$ 由 $(\boldsymbol{X}^{\mathrm{T}} \boldsymbol{Y} \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{X})_h$ 特征向量相应的最大特征值计算得到

$$\boldsymbol{w}_h \propto (\boldsymbol{X}^{\mathrm{T}} \boldsymbol{Y} \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{X})_h \boldsymbol{w}_h. \tag{7}$$

③PLS 的装载向量 $\boldsymbol{p}_h$ 和 $\boldsymbol{q}_h$ 计算如下:

$$\boldsymbol{p}_h^{\mathrm{T}}=\frac{\boldsymbol{w}_h^{\mathrm{T}}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})_h}{\boldsymbol{w}_h^{\mathrm{T}}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})_h\boldsymbol{w}_h},$$

$$\boldsymbol{q}_h^{\mathrm{T}}=\frac{\boldsymbol{w}_h^{\mathrm{T}}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y})_h}{\boldsymbol{w}_h^{\mathrm{T}}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})_h\boldsymbol{w}_h}.$$

④提取每个潜在向量运算后协方差矩阵 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 和 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}$ 的残差可以表示为

$$(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})_{h+1}=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})_h-\boldsymbol{p}_h\boldsymbol{p}_h^{\mathrm{T}}(\boldsymbol{t}_h^{\mathrm{T}}\boldsymbol{t}_h),$$

$$(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y})_{h+1}=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y})_h-\boldsymbol{p}_h\boldsymbol{q}_h^{\mathrm{T}}(\boldsymbol{t}_h^{\mathrm{T}}\boldsymbol{t}_h).$$

此外, A. Hoskuldsson[5]认为 $\boldsymbol{Y}$ 的退化不是必须的, 每次在计算协方差矩阵 $\boldsymbol{X}_h^{\mathrm{T}}\boldsymbol{Y}_h^{\mathrm{T}}\boldsymbol{Y}_h\boldsymbol{X}_h$ 特征向量 $\boldsymbol{w}$ 时, 发现 $\boldsymbol{X}_h^{\mathrm{T}}\boldsymbol{Y}_h^{\mathrm{T}}\boldsymbol{Y}_h\boldsymbol{X}_h$ 与 $\boldsymbol{X}_h^{\mathrm{T}}\boldsymbol{Y}_1^{\mathrm{T}}\boldsymbol{Y}_1\boldsymbol{X}_h$ 等价. 据此, M. Tenenhaus[7]、F. Lindgren[10]、De Jong[14]和 Zhun Yunhua[15]等提出 SIMPLS 算法, 其数学模型为

$$\max_{\|\boldsymbol{w}\|=1}\sum_{j=1}^{q}\mathrm{cov}(\boldsymbol{Y},\boldsymbol{X}_h\boldsymbol{w}_h).$$

同理, J. Hinkle[16]、M. Stone[17]和赵仕健[18]等证明 $\boldsymbol{X}$ 矩阵的退化也可以省略. 这时, PLS 回归算法的数学模型可以表示为

$$\arg\max_{\substack{\boldsymbol{w}\in\boldsymbol{R}^p,\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}=1\\ \boldsymbol{w}^{\mathrm{T}}\boldsymbol{R}_X\boldsymbol{w}_k=0^{\mathrm{T}}}}\left\{\sum_{i=1}^{q}<\boldsymbol{X}\boldsymbol{w},\boldsymbol{y}_i>^2\right\}.$$

此算法的优点是: 成分 $\boldsymbol{t}_i=\boldsymbol{X}\boldsymbol{w}_i$ 直接与初始的 $\boldsymbol{X}$ 而非与退化后的 $\boldsymbol{X}_i=\boldsymbol{X}_{i-1}-\boldsymbol{t}_{i-1}\boldsymbol{p}_{i-1}^{\mathrm{T}}$ 相联系[3], 所得到的结果更易于解释. 求解 $\boldsymbol{W}$ 过程中不涉及变量 $\boldsymbol{X}$ 与 $\boldsymbol{Y}$ 对所求得的成分 $\boldsymbol{t}_i=\boldsymbol{X}\boldsymbol{w}_i$ 的回归, 节省了对 $\boldsymbol{c}_h,\boldsymbol{u}_h,$和$\boldsymbol{F}_h$ 的计算, 计算过程得到了很大程度的简化, 计算更为简单.

2 kernel 算法的改进

算法的选择与待解决的问题类型高度相关. 所有 PLS 算法的优化都是针对不同类型的问题进行的, 没有哪一种算法适合解决所有问题. 在 QSAR 问题(如中药复方药代动力学研究)、自适应过程控制和校准更新等领域, 数据定期被收集更新, 解释矩阵和应用矩阵数据量非常大. 数据的特点是样本数远小于解释变量($N\ll K$)或者解释变量数远小于样本点数($K\ll N$). 对于这类建模问题, 由于模型的回归系数多, 解释变量或因子的重要性分析十分困难. 为适应在线系统实时快速建模的需求, 以及时、有效处理系统建模中涉及的大量数据, 减少计算所用的时间, 同时降低对计算机内存的需求, 需要有一种新的方法基于 kernel 算法和 SIMPLS 算法的加权递归, PLS 算法正是为解决这类问题而提出的. 该方法有效地利用计算结果进行递归计算, 实时地对新的模型参数进行修正, 从而得到理想的参数估计值, 以满足系统实时建模的要求.

B. hupinder [12]和王惠文等[8]证明得分矩阵 $\boldsymbol{T}$ 与解释矩阵 $\boldsymbol{X}$ 之间的关系可以用下列性质表示:

$$\boldsymbol{t}_h=\boldsymbol{X}\boldsymbol{R}=\boldsymbol{X}\prod_{j=1}^{h-1}(\boldsymbol{I}-\boldsymbol{w}_j\boldsymbol{p}_j^{\mathrm{T}})\boldsymbol{w}_h.$$

由文献[8]可知, 得分向量 $\boldsymbol{t}_h$ 和 $\boldsymbol{u}_h$ 与 $\boldsymbol{X}$ 矩阵及 $\boldsymbol{Y}$ 矩阵的轴 $\boldsymbol{w}_h$ 和 $\boldsymbol{c}_h$ 、载荷向量 $\boldsymbol{p}_h$ 和 $\boldsymbol{q}_h$ 及回归系数 $\boldsymbol{r}_i$ 之间存在循环计算关系, 只要知道其中之一, 就可以推知其它向量. 故

$$\begin{aligned}&\boldsymbol{r}_1=\boldsymbol{w}_1,\\&\boldsymbol{r}_i=\boldsymbol{w}_i-\boldsymbol{p}_1^{\mathrm{T}}\boldsymbol{w}_i\boldsymbol{r}_1-\boldsymbol{p}_2^{\mathrm{T}}\boldsymbol{w}_i\boldsymbol{r}_2-\cdots-\boldsymbol{p}_{i-1}^{\mathrm{T}}\boldsymbol{w}_i\boldsymbol{r}_{i-1},i>1.\end{aligned}\tag{8}$$

根据计算强度, 在 kernel 算法中决定计算速度的关键步骤是矩阵 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 和 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}$ 的构造. 由方程(8)可知, $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 的构造不是必须的. 如果 $\boldsymbol{X}$ 的样本点数远远大于其变量数($n\gg k$), 就可以很方便地计算 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$, 因为存储 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 比存储 $\boldsymbol{X}$ 占用更少的存储空间. 当不考虑计算机存储空间的限制时, 则可以不计算 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 而直接使用 $\mathbf{X}$ 计算 $\boldsymbol{p}$ 和 $\boldsymbol{q}$, 所需的计算量更少.

对于定期收集数据的任务(如 QSAR、自适应过程控制和校准更新等领域), 理想的做法是用每一个新的多变量对象(当它有效时)递归更新 PLS 模型. 这个过程可能是慢慢地随时间变化, 在指数加权方式中, 采取对最新数据加权和对旧的数据折扣的处理方式. 采用协方差更新方程的算法, 可以得到一个快速的递归指数加权更新的 PLS 回归模型的 kernel 算法

$$(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})_{h+1}=\lambda_t(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})_h+\boldsymbol{x}_t^{\mathrm{T}}\boldsymbol{x}_t,\tag{9}$$

$$(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y})_{h+1}=\lambda_t(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y})_h+\boldsymbol{x}_t^{\mathrm{T}}\boldsymbol{y}_t,\tag{10}$$

这里 $\boldsymbol{x}_t$ 及 $\boldsymbol{y}_t$ 是在时间 t 观测到的的新对象向量, $(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})_{h+1}$和$(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y})_{h+1}$是在 t 时刻更新的协方差矩阵. 在每一个新的采样周期, 协方差矩阵中以前的数据以遗忘因子 $\lambda_t(0<\lambda_t\leqslant 1)$ 倍乘旧数据, 新数据被添加到协方差矩阵中. 当 $\lambda_t=1$ 表示不需要对旧数据折扣. 由于在方程(9)和方程(10)中的协方差矩阵被用于更新的计算工作量很少, 因此, 本算法在这些应用中是非常快的. B.S.Dayal 等[26-27]将此方法应用于模拟连续搅拌槽反应器的自适应多变量控制和更新工业矿物浮选回路在线的多路输出的预测模型.

以下是加权递归 PLS 算法的描述:

① 计算协方差矩阵 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$, 这一步是可选择的.

② 如果 $\boldsymbol{Y}$ 的变量数较少，计算 $\boldsymbol{q}_h$ 作为 $(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{Y})_h$ 的最大特征值对应的特征向量，$\boldsymbol{w}_h$ 可以按如下关系式计算

$$\boldsymbol{w}_h^{\mathrm{T}} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y})_h\boldsymbol{q}_h,$$
$$\boldsymbol{w}_h = \boldsymbol{w}_h\big|\boldsymbol{w}_h\big|.$$

如果 $\boldsymbol{X}$ 的变量数较少，计算 $\boldsymbol{w}_h$ 作为 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{X}$ 的最大特征值对应的特征向量，如方程(7)所示.

③ 计算 $\boldsymbol{r}_h$ 的式子为

$$\boldsymbol{r}_1 = \boldsymbol{w}_1,$$
$$\boldsymbol{r}_h = \boldsymbol{w}_h - \boldsymbol{p}_1^{\mathrm{T}}\boldsymbol{w}_h\boldsymbol{r}_1 - \boldsymbol{p}_2^{\mathrm{T}}\boldsymbol{w}_h\boldsymbol{r}_2 - \cdots - \boldsymbol{p}_{h-1}^{\mathrm{T}}\boldsymbol{w}_h\boldsymbol{r}_{h-1}, \quad h>1.$$

④计算装载向量 $\boldsymbol{p}_h$ 和 $\boldsymbol{q}_h$.

利用原始矩阵 $\boldsymbol{X}$, 只须计算一次即可构造协方差矩阵 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$，然后在所有维中应用

$$\boldsymbol{p}_h^{\mathrm{T}} = \boldsymbol{r}_h^{\mathrm{T}}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})/(\boldsymbol{r}_h^{\mathrm{T}}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})\boldsymbol{r}_h),$$
$$\boldsymbol{q}_h^{\mathrm{T}} = \boldsymbol{r}_h^{\mathrm{T}}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y})_h/(\boldsymbol{r}_h^{\boldsymbol{T}}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})\boldsymbol{r}_h).$$

⑤更新协方差矩阵 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}$：由方程(5)~(6)和方程(9)~(10)得

$$(1-\lambda_t)(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y})_h = \boldsymbol{x}_t^{\mathrm{T}}\boldsymbol{y}_t + \boldsymbol{p}_h\boldsymbol{q}_h^{\mathrm{T}}(\boldsymbol{t}_h^{\mathrm{T}}\boldsymbol{t}_h),$$
$$(1-\lambda_t)(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})_h = \boldsymbol{x}_t^{\mathrm{T}}\boldsymbol{x}_t + \boldsymbol{p}_h\boldsymbol{p}_h^{\boldsymbol{T}}(\boldsymbol{t}_h^{\mathrm{T}}\boldsymbol{t}_h).$$

⑥存储 $\boldsymbol{w}, \boldsymbol{p}, \boldsymbol{q}$ 和 $\boldsymbol{r}$ 到 $\boldsymbol{W}, \boldsymbol{P}, \boldsymbol{Q}$ 和 $\boldsymbol{R}$:

$$\boldsymbol{W}=[\boldsymbol{w}_1\,\boldsymbol{w}_2\cdots\boldsymbol{w}_m],\boldsymbol{P}=[\boldsymbol{p}_1\,\boldsymbol{p}_2\cdots\boldsymbol{p}_m],\boldsymbol{Q}=[\boldsymbol{q}_1\,\boldsymbol{q}_2\cdots\boldsymbol{q}_m],\boldsymbol{R}=[\boldsymbol{r}_1\,\boldsymbol{r}_2\cdots\boldsymbol{r}_m].$$

⑦如果 $PRESS_{(\boldsymbol{h})} - PRESS_{(\boldsymbol{h}-1)} < \varepsilon, \varepsilon$ 为预测精度，则转入第⑧步; 否则转入第②计算下一个潜变量.

⑧当计算潜向量时, PLS 模型的回归因子(系数)由 $\boldsymbol{B}_{\boldsymbol{PLS}} = \boldsymbol{R}\boldsymbol{Q}^{\mathrm{T}}$ 计算.

模拟实验采用 Matlab 环境，由随机函数 rand 产生 1 000 个采样点作为系统分析的试验数据，对比传统的 kernel 算法，改进后的加权递归算法的计算速度明显提高. 说明加权递归 PLS 算法适合应用于海量数据的实时建模分析过程.

3　讨论

每种新的 PLS 算法的提出相比较于典型 PLS 算法都具有革命性[19]. 要证明新算法比典型 PLS 算法更有用，必须要考虑重复建模过程，其中涉及如交叉验证[20-21]、变量选择[22]和样本数据集中数据缺损等问题. 交叉验证技术可以帮助在不知道潜变量数的情况下选择适当数量的潜变量，以保证所建立模型的准确性和计算的高效性. S. Rannar[24]等研究了 Lindgren 提出的 kernel 算法中数据缺损问题. 他们采用的方法是 EM 算法[25]. 所有这些关键措施都可以加速重复建模的速度[23]. 这些技术在上述文献中已详细叙述，在此不作讨论.

PLS 算法的改进必须考虑在重复建模过程中矩阵的退化运算. 希望无论样本数还是变量数的增减，都不需要重复计算方差/协方差以及伴随矩阵. 比如，在计算 $\boldsymbol{p}_h$ 和 $\boldsymbol{q}_h$ 时，显然 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 要比 $\boldsymbol{X}$ 矩阵规模小得多，而在任何交叉验证方法中，都存在重复计算潜变量的情况，因此可以考虑一次性构造 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 矩阵，而不去用 $\boldsymbol{X}$ 矩阵. 显然，加权递归 PLS 算法计算潜变量更快.

在具体应用中，为了减弱旧数据的影响，引入遗忘因子 $\lambda_t(0<\lambda_t\leqslant 1)$，使算法收敛的速度变快，提高递归算法的稳健性，使得估计参数更加合理. 标准化处理方法与批处理有所不同，因为实时性要求高，数据量大，不可能对新引入的变量数据进行方差归一化处理. 通常在事先确定整体数据的变化范围的基础上采用其它的处理方法[28].

4　结论

对 PLS 算法的所有改进，都是为了适应特定的任务的. 要选择一个好的算法，应用首先明白应用领域是什么？想要做什么？回答了这些问题，再来选择合适的算法. 比如，当样本数远远大于变量数，并且数据是实时更新的建模过程，可以选择加权递归 PLS 算法. 这时，协方差矩阵 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 只要被计算一次，然后可被用于后续所有的 $\boldsymbol{p}$ 和 $\boldsymbol{q}$ 的计算. 当解释矩阵规模较小时，可能直接用 $\boldsymbol{X}$ 矩阵来计算 $\boldsymbol{p}$ 和 $\boldsymbol{q}$. 总之，在 PLS 算法中，只要对 $\boldsymbol{X}$ 矩阵或 $\boldsymbol{Y}$ 矩阵之一进行退化. 新算法还提供了以递归方式更新 PLS 模型以及对旧数据的加权处理以削弱旧数据的影响. 比较新算法和传统的 kernel 算法的执行速度，新算法明显更快，在执行交叉验证和处理错误数据时，新算法显示一定的优越性.

5　参考文献

[1] 王惠文, 朱韵华. PLS 回归在消除多重共线性中的作用 [J]. 数理统计与管理, 1996, 15(6): 48-52.

[2] Wold S. Partial least squate in Ess [M]. New York: Wiley, 1985 : 81-591.

[3] Wold S, Algano C, Dunn M , et al.Pattern regression finding and using regularities in multivariate data [M]. Landon: Analysis Applied Science Publication, 1983.

[4] Wold S. Modeling data labels by principal component and PLS : class patterns and quantitative predictive relations [J]. Analysis, 1984, 12: 477-485.

[5] Hoskuldsson A. PLS regression methods [J]. Journal of Chemometrics, 1988, 2(3): 211-228.

[6] Geladi P, Qkoulaski B. Partial least squares regression: a tutorial [J]. Analytical Chemical data, 1986, 35: 1-17.

[7] Tenenhaus M, Gauchi J P, Menardo C. Regression PLS et application [J]. Revue de statistique appliquce 1995, 53(1): 7-63.

[8] 王惠文. 偏最小二乘回归方法及其应用 [M]. 北京: 国防工业出版社, 1999.

[9] Fornell C. A second genetation of multivarite analysis [M].New York: Pracger, 1982.

[10] Lindgren F, Geladi P, Wold S. The kernel algorithm for PLS [J]. J Chemometrics, 1993, 7: 45.

[11] Hoskuldsson A. A combined theory for PCA and PLS [J]. J Chemometrics, 1995, 9: 91.

[12] Hupinder B, Dayal S, John, Macgregor F. Improved PLS algorithms [J]. Journal of Chemometrics, 1997, 11: 73-85.

[13] Jong S D, Braak C J F T.JChemometrics, 1994, 8: 169.

[14] Jong S D. SIMPLS: an alternative approach to partial least squares regression [J]. Chemometrics Intelligent Laboratory Systems, 1993, 18: 251-263.

[15] Zhu Yunhua, Wang Huiwen, Yang Xianglong. A simplified algorithm of PLS regression [J]. Journal of Systems Science and Systems Engineering, 2000, 19(4): 414-419.

[16] Hinkle J, Rayens W. Partial least squares and compositional data: problens and alternatives [J]. Chemometrics and Intelligent Laboratory Systems, 1995, 30(1): 159-172.

[17] Stone M, Brooks K J. Continum regression: Cross-validation sequentially constructed prediction embracing ordinary least squares, partial least squares and principal components regression [J]. J the Royal statistical society series B(Methodological), 1990, 52(2): 237-269.

[18] 赵仕健, 徐用懋. 部分最小二乘算法的神经元网络实现 [J]. 清华大学学报: 自然科学版, 2004, 44(10): 1348-1351.

[19] Bush B L, Nachbar J R B. Sample-distamce partial least squares: PLS optimzed for many wariables, with application to CoMFA [J]. J Compute-Aided Mol, Design, 1993, 7: 587-619.

[20] StoneM. Cross-balidatory choice and assessment fo statistical predictions [J]. J Royal Stat, Soc B, 1974, 36: 111-133.

[21] Geisser S. A predictive approach to the random effect model [J]. Biometrika, 1974, 61: 101-107.

[22] Baroni M, Costantino G, Riganelli D, et al. Generationg optimal linear PLS estimations(GOLPE): an advanced chemometric tool for handing 3D QSAR problems [J]. Quant Struct-Act Relat, 1993, 12: 9-20.

[23] Fredrik L, Stefan R. Alternative partial least-squares(PLS) algorithms [J]. Perspectives in Drug Discovery and Design, 1998(12/14): 105-113, .

[24] Rannar S, Geladi P, Lindgren F, et al. A PLS kernel algorithm for data sets with mahy variables and less objects: part 2.Cross-validation, missing data and examples [J]. J Chemometrics, 1995, 9: 459-470.

[25] Little R J A, Rubin D B. Statistical analysis with missing data [M]. New York: Wiley, 1987.

[26] Dayal B S. Department of chemical engineering [D]. McMaster : McMaster University, 1996.

[27] Dayal B S, MacGregor J F. J Process Control, 1996.

[28] 刘强, 尹力. 一种简化递推偏最小二乘建模算法及其应用 [J]. 北京航空航天大学学报, 2003, 29(7): 640-643.

The Improved Partial Least Squares Regression Recommendation Algorithm

LIAO Chun-hua, DU Jian-qiang, CHEN Chun-lei, LI Zhi-biao

(School of Computer Science, Jiangxi University of Traditional Chinese Medicine, Nanchang Jiangxi 330004, China)

Abstract: Based on the related PLS algorithms, a new improved recursive exponentially weighted PLS regressions algorithms was derived for the QSAR research and industrial process control modeling. Simulation experiments show that in the real-time modeling process, the performance of this algorithm is superior to the traditional PLS regression algorithm.

Key words: partial least squares(PLS)regression; kernel algorithm; algorithms improvement; recursive exponentially weighted algorithms

(责任编辑: 冉小晓)