概率型稀疏核 Logistic 多元分类机

郑建炜* 王万良 蒋一波 陈伟杰
(浙江工业大学计算科学与技术学院 杭州 310023)

摘 要：该文提出一种基于二级先验概率的多元核 Logistic 分类机，扩展核 Logistic 回归为多元模型，并解决其解的稀疏性问题，以提升多分类应用时的模型运行速率。为约简模型构建所需计算量，训练过程采用自下向上增补算法，每次迭代采用尽量少的输入样本，规避了大型矩阵逆操作，以适应于不同量度的数据场合。实验显示，所提多元分类机模型构建简单，且识别率与稀疏性都优于经典支持向量机所生成的“一对一”多分类方法及传统多元核 Logistic 回归算法。

关键词：核 Logistic 回归；稀疏性概率；多元分类机；自下向上训练
中图分类号：TP391.4 **文献标识码**： A **文章编号**：1009-5896(2011)07-1632-07
DOI: 10.3724/SP.J.1146.2010.01237

Probabilistic Sparse Kernel Logistic Multi-classifier

Zheng Jian-wei Wang Wan-liang Jiang Yi-bo Chen Wei-jie
(*College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou* 310023, *China*)

Abstract: A new kernel logistic regression model based on two phase sparsity-promoting prior is proposed to render a sparse multi-classifier and enhance the run-time efficiency. For accelerating the building of the model, the bottom-up training algorithm is adopted which controls the capacity of the learned classifier by minimizing the number of basis functions used, resulting in better generalization and faster computation. Experimental results on standard benchmark data sets attest to the accuracy, sparsity, and efficiency of the proposed methods.

Key words: Kernel logistic regression; Sparsity-promoting prior; Multi-classifier; Bottom-up training

1 引言

支持向量机(Support Vector Machines, SVM)是一种经典的二元模型，已经成功地应用于不同分类领域，然而如何对它进行多元扩展却始终是一个难以解决的问题，一般都通过“一对一”，“一对多”等后处理技术间接实现多目标分类。除此之外，它的不足还表现在：二元输出结果不如概率输出贴近实际、支持向量数随样本量增加呈线性递增、核函数需严格满足 Mercer 条件、模型泛化参数 C 与核函数参数 σ 都需交叉验证确定致使训练时间过长等。相关向量机[1](Relevance Vector Machines, RVM)虽然弥补了 SVM 的众多缺陷，但其仍是一个二元模型，不能直接应用于多分类任务。

核 Logistic 回归模型是另一种强大的有监督非线性分类机，在无线传感网可靠性评估[2]、图像分割[3]、连续语音识别[4]等方面都有不俗的表现。KLR 的核函数无需满足 Mercer 条件，其概率输出结果给出了最终类别赋值的置信度，符合众多分类任务的实际要求，更加自然且人性化。实际分类问题如人脸识别、说话人辨认、文本分类等都具有多个识别目标，传统 KLR 为二元分类机，应用时往往要先进行多分类扩展。Roth[5]将 KLR 依成对耦合方式实现多目标分类，与 SVM “一对一” (SVM One Versus One, SVMOVO)策略一致，虽然功能上实现了多目标分类，但其本质仍是若干个二元分类器的组合。Karsmakers 等人[6]从模型本身出发，提出真正的多元核 Logistic 回归模型(Multi-class Kernel Logistic Regression, MKLR)，并用正则化迭代重加权最小平方算法进行参数优化，迭代过程要进行矩阵逆操作，因此训练效率较低，特别不适于大数据样本场合，且模型不具有稀疏性。

稀疏性是 SVM 与 RVM 都具有的优秀特性，能防止模型参数过学习，提升模型泛化能力，且能更好地满足模型应用时的实时性。文献[7]将原输入特征向量映射到一个子空间，以特征向量选择的方式实现了 KLR 的稀疏性，与 IVM(Import Vector Machines)[8]原理如出一辙。Krishnapuram[9]为模型参数绑定 Laplasian 先验概率、Fu 等人[10]则将模型参数叠加 $L_{1,2}$ 惩罚因子，两者都从本质上实现了 KLR 模型的稀疏性，但同时却又引入了一个惩罚平

2010-11-15 收到，2011-03-24 改回
国家自然科学基金(61070043)和浙江省自然科学基金(Y1100611)资助课题
*通信作者：郑建炜 weibaby2007@163.com

衡因子系数，需要通过交叉验证搜索获取，极大地增加了训练负担。

本文旨在为多元模型 MKLR 添加稀疏性，同时提出相应的模型训练算法，本文结构为：第 2 节介绍了扩展的多元核 Logistic 回归模型，并在不增加新参数的前提下为模型参数绑定二级先验概率，引入稀疏性；为使模型能够应用于不同训练样本数量情形，第 3 节采用自下向上的模型训练算法对参数进行交替迭代优化；第 4 节以基准公用数据对模型效率进行了验证；第 5 节为本文总结。

2　稀疏性多元核 Logistic 分类机

2.1 多元核 Logistic 分类

有监督 K 类分辨任务中，给定训练样本集 $\{\boldsymbol{x}_n,\boldsymbol{t}_n\}_{n=1}^N$，其中 $\boldsymbol{x}_n$ 为 p 维输入特征向量，即 $\boldsymbol{x}_n=(x_{n1},x_{n2},\cdots,x_{np})^{\mathrm{T}}$；$\boldsymbol{t}_n$ 为 K 维输出向量。当输入 $\boldsymbol{x}_n$ 属于类别 $k\in\{1,2,\cdots,K\}$ 时，则 $t_{nk}=1$；反之，$t_{nk}=0$。训练目标为：最优化判别模型，使得给定新的输入特征向量 $\boldsymbol{x}$ 时，分类机能够从 $\{1,2,\cdots K\}$ 中选取一个正确的类别标签，使得 $t_k=1$，任意 $t_{i\neq k}=0$。

多元 Logistic 分类模型(Multi-class Logistic Classifier, MLC)中，给定最优化模型参数为 $\boldsymbol{\omega}\in\mathbb{R}^{K\times p}$，则当输入 $\boldsymbol{x}$ 时，结果输出为 k 的概率计算如下：

$$P(t_k=1\mid\boldsymbol{x},\boldsymbol{\omega})=\frac{\exp(\boldsymbol{\omega}_k^{\mathrm{T}}\boldsymbol{x})}{\sum_{j=1}^{K}\exp(\boldsymbol{\omega}_j^{\mathrm{T}}\boldsymbol{x})} \tag{1}$$

其中 $\boldsymbol{\omega}_k$ 是类别 k 的权值向量。由于各类概率输出之和为 1，可将目标多元函数输出概率改成[6]：

$$P(t_k=1\mid\boldsymbol{x},\boldsymbol{\omega})=\begin{cases}\dfrac{\exp(\boldsymbol{\omega}_k^{\mathrm{T}}\boldsymbol{x})}{1+\sum_{j=1}^{K-1}\exp(\boldsymbol{\omega}_k^{\mathrm{T}}\boldsymbol{x})}, & k\in 1,\cdots,K-1\\ \dfrac{1}{1+\sum_{j=1}^{K-1}\exp(\boldsymbol{\omega}_k^{\mathrm{T}}\boldsymbol{x})}, & k=K\end{cases} \tag{2}$$

经调整后模型判别结果不会受到影响，参数 $\boldsymbol{\omega}$ 只需训练 $\{\boldsymbol{\omega}_k\}_{k=1}^{K-1}$ 即可，能提升模型训练速率。在有监督学习情况下，给定 N 个训练样本，模型参数 $\{\boldsymbol{\omega}_k\}_{k=1}^{K-1}$ 的最大似然估计可通过最大化如下 lg 似然函数获取：

$$\begin{aligned}&\lg P(\boldsymbol{t}|\boldsymbol{X},\boldsymbol{\omega})\\&=\lg\prod_{n=1}^{N}\prod_{k=1}^{K-1}(p(t_{nk}=1\mid\boldsymbol{x}_n,\boldsymbol{\omega}))^{t_{nk}}\\&=\sum_{n=1}^{N}\left\{\sum_{k=1}^{K-1}t_{nk}\boldsymbol{\omega}_k^{\mathrm{T}}\boldsymbol{x}_n-\lg\left(1+\sum_{j=1}^{K-1}\exp(\boldsymbol{\omega}_j^{\mathrm{T}}\boldsymbol{x}_n)\right)\right\}\end{aligned} \tag{3}$$

式(2)，式(3)分别为多元 Logistic 分类机的决策目标与参数优化泛函，为增强模型非线性分类能力，将核技巧应用于原输入特征空间，则相应的多元核 Logistic 分类机(MKLC)参数改成 $\boldsymbol{\omega}\in\mathbb{R}^{K\times N}$，而优化泛函变成：

$$\begin{aligned}L&=\lg P(\boldsymbol{t}\mid\boldsymbol{X},\boldsymbol{\omega})\\&=\sum_{n=1}^{N}\left\{\sum_{k=1}^{K-1}t_{nk}\boldsymbol{\omega}_k^{\mathrm{T}}\varphi(\boldsymbol{x}_n)-\lg\left(1+\sum_{j=1}^{K-1}\exp(\boldsymbol{\omega}_j^{\mathrm{T}}\varphi(\boldsymbol{x}_n))\right)\right\}\end{aligned} \tag{4}$$

其中 $\varphi(\boldsymbol{x}_n)=\{k(\boldsymbol{x}_n,\boldsymbol{x}_1),\cdots,k(\boldsymbol{x}_n,\boldsymbol{x}_N)\}$ 是样本 $\boldsymbol{x}_n$ 的核函数或基函数列，应用最为广泛的核函数为径向基核：$K(x,y)=\exp(-\|x-y\|^2/2\sigma^2)$。相应的 MKLC 目标决策函数改为

$$\max_{k\in\{1,2,\cdots,K\}}\left\{\max\left\{\frac{\exp(\boldsymbol{\omega}_k^{\mathrm{T}}\varphi(\boldsymbol{x}))}{1+\sum_{j=1}^{K-1}\exp(\boldsymbol{\omega}_j^{\mathrm{T}}\varphi(\boldsymbol{x}))}\right\}_{k=1}^{K-1},\ \frac{1}{1+\sum_{j=1}^{K-1}\exp(\boldsymbol{\omega}_j^{\mathrm{T}}\varphi(\boldsymbol{x}))}\right\} \tag{5}$$

2.2 稀疏性先验概率

为防止模型训练过学习，控制模型复杂度，一般采取参数 $\boldsymbol{\omega}$ 叠加 L_2 惩罚因子 $\frac{\lambda}{2}\sum_{k=1}^{K-1}\|\boldsymbol{\omega}_k\|_2$ 的方法，如支持向量机；引入 L_2 惩罚因子的不足是相应模型不具备稀疏性，如果将其改成 L_1 范数 $\lambda\sum_{k=1}^{K-1}\|\boldsymbol{\omega}_k\|_1$ [11]，则模型具有稀疏性特征，即训练结果只有部分参数取有意义的值，其余为零，使模型运行效率提升，但是新增的系数 λ 致使模型训练负担增加，需要交叉验证过程才能确定其最终取值，延缓了模型的优化过程；另一种稀疏性方法为模型参数绑定拉普拉斯先验概率[12]为

$$\begin{aligned}P(\boldsymbol{\omega}\mid\lambda)&=\prod_{k=1}^{K-1}\prod_{n=1}^{N}\frac{\lambda}{2}\exp(-\lambda\mid\omega_{kn}\mid)\\&=(\lambda/2)^{N\times(K-1)}\exp\left(-\lambda\sum_{k=1}^{K-1}\|\boldsymbol{\omega}_k\|_1\right)\end{aligned} \tag{6}$$

很明显，Laplace 先验与 L_1 惩罚因子一致，也需要进行 λ 参数的交叉验证取值过程。为弥补已有方法的缺陷，本文引入稀疏贝叶斯框架[13,14]中为参数添加两级先验分布的思想实现 MKLR 的稀疏性。

首先设定参数 ω_{ki} 服从均值为 0 方差为 α_i^{-2} 的高斯分布，即

$$P(\boldsymbol{\omega}_k\mid\boldsymbol{\alpha})=\prod_{i=1}^{N}N(\omega_{ki}\mid 0,\alpha_i^{-2}) \tag{7}$$

其中 $\boldsymbol{\omega}_k$ 为第 k 类模型参数，N 维向量 $\boldsymbol{\alpha}$ 是决定模型参数 $\boldsymbol{\omega}_k$ 的超参数。引入高斯先验分布可使模型计算可行，却无法产生稀疏性结果，因此限定超参数 $\boldsymbol{\alpha}$ 的超先验分布为

$$P(\boldsymbol{\alpha})=\prod_{i=1}^{N}\mathrm{Gamma}(\alpha_i\mid a,b) \tag{8}$$

其中 $\mathrm{Gamma}(\alpha_i\mid a,b)=\Gamma(a)^{-1}b^{a}\alpha_i^{a-1}e^{-b\alpha_i}$，$\Gamma(a)=\int_0^{\infty}t^{a-1}e^{-t}\mathrm{d}t$，为了达到无信息先验假设，一般取 $a=b=10^{-4}$。

这样，每一个 $\alpha_i>0$ 独立控制相应核函数对最终优化模型的相关度，这也是导致模型具有稀疏性的根本原因。为区别于 RVM 中的相关向量，在 α_i 值趋于无穷大时，相应 $\boldsymbol{\omega}$ 模型参数值趋向于零，称对应样本为无效向量；反之，则相应参数取有意义的数值，对应样本称为有效向量。

3 模型训练算法

3.1 算法整体框架

添加上述两级先验分布于模型参数后，最终概率型稀疏核 Logistic 多元分类机(Sparse MKLC, SMKLC)目标优化泛函由式(4)改为

$$\begin{aligned}\tilde{L}&=\lg\{P(\boldsymbol{t}\mid\boldsymbol{X},\boldsymbol{\omega})P(\boldsymbol{\omega}\mid\boldsymbol{\alpha})P(\boldsymbol{\alpha})\}\\&=\sum_{n=1}^{N}\left\{\sum_{k=1}^{K-1}t_{nk}\boldsymbol{\omega}_k^{\mathrm{T}}\varphi(\boldsymbol{x}_n)-\lg\left(1+\sum_{j=1}^{K-1}\exp(\boldsymbol{\omega}_j^{\mathrm{T}}\varphi(\boldsymbol{x}_n))\right)\right\}\\&\quad-\frac{1}{2}\sum_{k=1}^{K-1}\boldsymbol{\omega}_k^{\mathrm{T}}\boldsymbol{\alpha}\boldsymbol{\omega}_k+\sum_{n=1}^{N}(a\lg\alpha_i+b\alpha_i)\end{aligned} \tag{9}$$

由于 $a=b=10^{-4}$，式(9)最后部分在具体求解过程中可忽略。目标泛函 $\tilde{L}$ 的优化对象是模型参数 $\boldsymbol{\omega}$，而 $\boldsymbol{\omega}$ 的确定首先需要获取合理的超参数，因此训练过程分为 $\boldsymbol{\omega}$ 与 $\boldsymbol{\alpha}$ 交替迭代更新两个阶段[15]：

(1)给定超参数的前提下，最大化参数 $\boldsymbol{\omega}$ 后验概率，由于 $P(\boldsymbol{\omega}\mid\boldsymbol{t},\boldsymbol{\alpha})\propto P(\boldsymbol{t}\mid\boldsymbol{\omega})P(\boldsymbol{\omega}\mid\boldsymbol{\alpha})$，因此获取最优化 $\hat{\boldsymbol{\omega}}$ 等同于最大化目标泛函 $\tilde{L}$。

$$\hat{\boldsymbol{\omega}}=\{\hat{\omega}_k\}_{k=1}^{K-1}=\mathrm{IRLS}\max_{\boldsymbol{\omega}}\{\tilde{L}\} \tag{10}$$

其中 IRLS 代表最优化过程采用经典的迭代重加权最小平方算法[16]。

(2)计算目标泛函式(9)的 Hess 矩阵

$$\boldsymbol{H}=\nabla_{\boldsymbol{\omega}}\nabla_{\boldsymbol{\omega}}\lg P(\boldsymbol{\omega}\mid\boldsymbol{t},\boldsymbol{\alpha})\mid_{\hat{\boldsymbol{\omega}}}=-(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{V}_k\boldsymbol{\Phi}+\boldsymbol{A}) \tag{11}$$

其中 $\boldsymbol{A}=\mathrm{diag}(\alpha_1^{-2},\alpha_2^{-2},\cdots,\alpha_N^{-2})$，应用 Laplace 方法[17]将参数 $\boldsymbol{\omega}$ 后验概率 $P(\boldsymbol{\omega}\mid\boldsymbol{t},\boldsymbol{\alpha})$ 近似成高斯分布，其中高斯中心等于 $\hat{\boldsymbol{\omega}}$，协方差矩阵通过式(11)计算：

$$\boldsymbol{\Sigma}_k=(-\boldsymbol{H}_{\hat{\boldsymbol{\omega}}})^{-1}=(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{V}_k\boldsymbol{\Phi}+\boldsymbol{A})^{-1} \tag{12}$$

其中对角矩阵 $\boldsymbol{V}_k=\mathrm{diag}(\nu_{k1},\nu_{k2},\cdots,\nu_{kN})$，元素 $\nu_{kn}=P(t_{kn}\mid\boldsymbol{X},\boldsymbol{\omega})(1-P(t_{kn}\mid\boldsymbol{X},\boldsymbol{\omega}))$。根据优化所得 $\hat{\boldsymbol{\omega}}$，模型当前的训练结果为

$$\hat{\boldsymbol{t}}_k=\boldsymbol{\Phi}\hat{\boldsymbol{\omega}}+\boldsymbol{V}_k^{-1}(\boldsymbol{t}_k-P(\boldsymbol{t}_k\mid\boldsymbol{X},\boldsymbol{\omega})) \tag{13}$$

(3)根据所生成的 $\hat{\boldsymbol{\omega}}$，采用最大化边缘似然函数进行超参数 $\boldsymbol{\alpha}$ 的迭代更新，即最大化。

$$L(\boldsymbol{\alpha})=\lg P(\boldsymbol{t}\mid\boldsymbol{\alpha})=\lg\int_{-\infty}^{\infty}P(\boldsymbol{t}\mid\boldsymbol{\omega})P(\boldsymbol{\omega}\mid\boldsymbol{\alpha})\mathrm{d}\boldsymbol{\omega} \tag{14}$$

其中 $P(\boldsymbol{\omega}\mid\boldsymbol{\alpha})$ 采用第 2 步的近似高斯函数。由于 $P(\boldsymbol{t}\mid\boldsymbol{\omega})$ 的存在，对式(14)直接积分不能成功，需要依式(13)把 $\hat{\boldsymbol{t}}_k$ 看作是以 $\boldsymbol{\Phi}\hat{\boldsymbol{\omega}}$ 为中心，$\boldsymbol{V}^{-1}$ 为方差的高斯函数，再由式(14)计算出 $L(\boldsymbol{\alpha})$ 的优化函数：

$$\begin{aligned}L(\boldsymbol{\alpha})&=\lg\prod_{k=1}^{K-1}\mid\boldsymbol{M}_k\mid^{-1/2}\exp(-(1/2)\boldsymbol{t}_k^{\mathrm{T}}\boldsymbol{M}_k^{-1}\boldsymbol{t}_k)\\&=-\sum_{k=1}^{K-1}\{\lg\mid\boldsymbol{M}_k\mid+\boldsymbol{t}_k^{\mathrm{T}}\boldsymbol{M}_k^{-1}\boldsymbol{t}_k\}\end{aligned} \tag{15}$$

其中

$$\boldsymbol{M}_k=\boldsymbol{V}_k+\boldsymbol{\Phi}\boldsymbol{A}^{-1}\boldsymbol{\Phi}^{\mathrm{T}} \tag{16}$$

3.2 超参数优化

通过边缘最大似然式(15)对超参数 $\boldsymbol{\alpha}$ 进行优化可以采用两种不同的方法，即自上向下削减模型和自下向上增补模型。前者是将 $\boldsymbol{\alpha}$ 所有元素初始化为有效值，通过迭代算法，$\boldsymbol{\alpha}$ 中部分元素值趋向于无穷大，则 $\boldsymbol{\omega}$ 中相应值趋向于零，即削减有效向量至模型收敛。后者是 $\boldsymbol{\alpha}$ 中任取若干个元素为有效值，其余初始化为无穷大，通过迭代算法，逐渐地叠加、重估或删除有效向量，即增补模型。由于在训练样本较大时，自上向下的方法在初始阶段往往计算量过大或内存溢出，本文采用自下向上的方法。

式(16)的 $\boldsymbol{M}_k$ 中包含完整的 $\boldsymbol{\alpha}$ 向量，为适应自下向上的增补方法，需要先分解 $\boldsymbol{M}_k$，独立体现 α_i 的作用。

$$\begin{aligned}\boldsymbol{M}_k&=\boldsymbol{V}_k+\sum_{j\neq i}\alpha_j^{-1}\boldsymbol{\varphi}_j\boldsymbol{\varphi}_j^{\mathrm{T}}+\alpha_i^{-1}\boldsymbol{\varphi}_i\boldsymbol{\varphi}_i^{\mathrm{T}}\\&=\boldsymbol{M}_{k(-i)}+\alpha_i^{-1}\boldsymbol{\varphi}_i\boldsymbol{\varphi}_i^{\mathrm{T}}\end{aligned} \tag{17}$$

其中 $\boldsymbol{M}_{k(-i)}$ 为从 $\boldsymbol{M}_k$ 中把 ϕ_i 剥离后的矩阵。再应用矩阵行列式与逆矩阵定义可得

$$\mid\boldsymbol{M}_k\mid=\mid\boldsymbol{M}_{k(-i)}\mid\mid1+\alpha_i^{-1}\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{M}_{k(-i)}^{-1}\boldsymbol{\varphi}_i\mid \tag{18}$$

$$\boldsymbol{M}_k^{-1}=\boldsymbol{M}_{k(-i)}^{-1}-\frac{\boldsymbol{M}_{k(-i)}^{-1}\boldsymbol{\varphi}_i\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{M}_{k(-i)}^{-1}}{\alpha_i+\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{M}_{k(-i)}^{-1}\boldsymbol{\varphi}_i} \tag{19}$$

由此，式(15)可以转换成

$$L(\boldsymbol{\alpha})=-\sum_{k=1}^{K-1}\{\lg|\boldsymbol{M}_{k(-i)}|+\boldsymbol{t}_k^{\mathrm{T}}\boldsymbol{M}_{k(-i)}^{-1}\boldsymbol{t}_k\}+\sum_{k=1}^{K-1}\left\{\lg\alpha_i-\lg(\alpha_i+\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{M}_{k(-i)}^{-1}\boldsymbol{\varphi}_i)+\frac{(\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{M}_{k(-i)}^{-1}\boldsymbol{t}_k)^2}{\alpha_i+\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{M}_{k(-i)}^{-1}\boldsymbol{\varphi}_i}\right\}=L(\alpha_{-i})+\sum_{k=1}^{K-1}\left\{\lg\alpha_i-\lg(\alpha_i+s_{ki})+\frac{q_{ki}^2}{\alpha_{ki}+s_{ki}}\right\}=L(\alpha_{-i})+l(\alpha_i)\tag{20}$$

其中 $s_{ki}=\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{M}_{k(-i)}^{-1}\boldsymbol{\varphi}_i$，$q_{ki}=\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{M}_{k(-i)}^{-1}\boldsymbol{t}_k$。通过对式(20)一阶微分获取 $L(\boldsymbol{\alpha})$ 针对 α_i 的最优值。

$$\frac{\partial L(\boldsymbol{\alpha})}{\partial\alpha_i}=\sum_{k=1}^{K-1}\left\{\frac{1}{\alpha_i}-\frac{1}{\alpha_i+s_{ki}}-\frac{q_{ki}^2}{(\alpha_i+s_{ki})^2}\right\}=0\tag{21}$$

通过求解以上多项式，可以计算出相应 α_i 的稳定值，同时考虑 $\alpha_i=\infty$ 的情况，即可获得最优的 α_i。

$$\alpha_i^{\mathrm{opt}}=\arg\max_{\alpha_i}l(\alpha_i)\tag{22}$$

值得一提的是，上述最优值结果如为 $\alpha_i=\infty$，即相应训练样本为无效向量，由此产生了模型稀疏性。为简化 s_{ki} 及 q_{ki} 的计算，我们求助于因子 S_{ki}，Q_{ki}：

$$S_{ki}=\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{M}_k^{-1}\boldsymbol{\varphi}_i=V_{ki}\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{\varphi}_i-V_{ki}^2\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\Sigma}_k\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\varphi}_i\tag{23}$$

$$Q_{ki}=\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{M}_k^{-1}\boldsymbol{t}_k=V_{ki}\boldsymbol{\varphi}_i^{\mathrm{T}}t_k-V_{ki}^2\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\Sigma}_k\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{t}_k\tag{24}$$

使得

$$s_{ki}=\frac{\alpha_{ki}S_{ki}}{\alpha_{ki}-S_{ki}}\tag{25}$$

$$q_{ki}=\frac{\alpha_{ki}Q_{ki}}{\alpha_{ki}-S_{ki}}\tag{26}$$

从式(23)，式(24)可见，S_{ki}，Q_{ki} 的计算过程中不需要将第 i 个基函数 $\boldsymbol{\varphi}_i$ 去除，且其中的 $\boldsymbol{\Phi}$，$\boldsymbol{\Sigma}_k$ 仅需要当前所包括的有限个基函数元素，而非所有 N 个基函数。因此可提升 s_{ki}，q_{ki} 的求取速率。

综上所述，在自下向上的先验 $\boldsymbol{\alpha}$ 训练算法中，每次迭代过程即遍历所有基函数 $\boldsymbol{\varphi}_m$，找出其中使 $L(\boldsymbol{\alpha})$ 增加最大的相应 α_i^{opt}，满足

$$m=\arg\max_i(L(\alpha_i))\tag{27}$$

设定所搜得的基函数 $\boldsymbol{\varphi}_m$ 原先验系数为 α_m^{old}，根据不同的情形对搜索到的 α_m^{old} 进行调整：

(1)如果 $\alpha_m^{\mathrm{old}}=\infty$ 且 $\alpha_m^{\mathrm{opt}}<\infty$，则当前模型没有包括基函数 $\boldsymbol{\varphi}_m$，且 $\boldsymbol{\varphi}_m$ 为有效向量，进行叠加操作将之包含进模型，且依式(22)修正 α_m^{old}；

(2)如果 $\alpha_m^{\mathrm{old}}<\infty$ 且 $\alpha_m^{\mathrm{opt}}=\infty$，则当前模型已包括基函数 $\boldsymbol{\varphi}_m$，且 $\boldsymbol{\varphi}_m$ 为无效向量，进行删除操作将之从模型中剔除，且将 α_m^{old} 赋值为无穷大；

(3)如果 $\alpha_m^{\mathrm{old}}<\infty$ 且 $\alpha_m^{\mathrm{opt}}<\infty$，则当前模型已包括基函数 $\boldsymbol{\varphi}_m$，即 $\boldsymbol{\varphi}_m$ 仍为有效向量，进行重估操作，依式(22)修正 α_m^{old}。

3.3 实现细则

综合上节分析内容，所述的自下向上模型训练算法具体步骤如下：

(1)设定任意 p 个基函数 $\boldsymbol{\varphi}_i$ 为初始模型元素，相应 $\alpha_i=1$。如果 p 值选取过大，则会影响模型初始阶段训练速度，违背了自下向上算法的基本思想，多次实验显示任意选择 $p\in\{1,\cdots,10\}$，模型结果相差无几；

(2)通过 Laplace 近似过程计算当前模型的 $\boldsymbol{\Sigma}$、$\boldsymbol{\omega}$ 及每个基函数 $\boldsymbol{\varphi}_m$ 对应的 $s_{km}\,|_{k=1}^{K-1}$，$q_{km}\,|_{k=1}^{K-1}$，为模型迭代更新做好准备；

(3)依式(27)从 N 个基函数中选择使 $L(\boldsymbol{\alpha})$ 增益最大的 $\boldsymbol{\varphi}_m$。然而，当训练样本数 N 过大时，每次迭代过程中 $\boldsymbol{\varphi}_m$ 的搜索占据了大量时间，可采取的方法是随机选择 $x\ll N$ 个基函数，从中选取最佳的基函数。另外，每次迭代中只进行一个基函数的修整过于浪费系统资源，可对基函数进行增益排序，每次迭代取前 d 个基函数进行修整。本文采用 $x=100$，$d=5$；

(4)依式(22)求取 $\alpha_i^{\mathrm{opt}}\,|_{i=1}^{d}$，并根据 3.2 节所述的 3 种不同情形修整模型；

(5)迭代更新修整后模型的 $\boldsymbol{\Sigma}$，$\boldsymbol{\omega}$，计算新选取的 x 个基函数相应 $s_{ki}\,|_{i=1}^{x}$，$q_{ki}\,|_{i=1}^{x}$；

(6)当模型不再有基函数叠加、删除操作，且重估操作中 $\max|\Delta\alpha_m|<10^{-4}$，则模型收敛；反之，转到步骤 3 继续训练过程。

4 实验分析

为验证所提稀疏核 Logistic 多元分类器的算法效率及分类性能，分别采用人造数值与 UCI 公共测试数据库中的部分数据集进行直观分类显示与不同算法间的性能对比。算法所需核函数都采用径向基核，其中核带宽 σ 由 5-fold 交叉验证及网格式搜索过程获取，所有实验结果都在 CPU 主频为 2.6 GHz，内存 4G 的台式机中产生，软件环境为 Windows XP 操作系统，Matlab7.1 编译平台。

4.1 合成数值

首先在平面上以坐标[0,1]为区间随机产生 3 类 2 维数据，每类数据样本量为 50，不同类别样本间存在交叉区域，应用 SMKLC 后的分类效果如图 1 所示，其中加圆圈的数据为权值参数不为零的特征向量，即有效向量。

从图 1 中可见，对于给定的合成三分类数据，通过 SMKLC 训练所得的决策边界为样本提供了很好的区分性。除此之外，上述 SMKLC 的最终有效

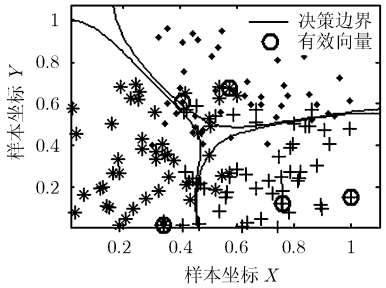

图 1 SMKLC 分类合成数据效果直观图

向量数仅为 5 个，而完整的训练样本数为 150 个，说明其具有很强的稀疏性能，在应用于多分类数据时，保证了算法运行的高实时性能。

4.2 UCI 基准数据实验 1

继续验证 SMKLC 的具体识别效率，选取公共测试数据库 UCI 上的数据集 Iris, Wine, Libras, Svmguide2, Waveform 分别进行测试，各数据描述如表 1 所示，其中 NTR 表示训练样本数量，NTE 代表测试样本数量，Dims 和 Class 分别是样本特征维度及数据类别总数。

表 1 UCI 5 种数据描述

UCI Data	NTR	NTE	Dims	Class
Iris	120	60	4	3
Wine	146	178	13	3
Libras	360	270	90	15
Svmguide2	150	391	20	3
Waveform	2000	2000	21	3

在测试过程中，实现了经典 SVMOVO, RVMOVO 多分类方法以及文献[6]多元核 Logistic 方法(MKLR)与 SMKLC 进行对比，结果如表 2 所示，每种算法都列出了所有数据的识别错误率(ER)以及各自的平均相关向量数(RV)、支持向量数(SV)和有效向量数(UV)。从中可见，RVM 和 SVM 两者在不同数据类型下识别率相近，且 RVM 具有更高的稀疏性，RV 的数量较 SV 明显减少，两者的缺陷是都通过 OVO 方式实现多分类扩展，使得模型构建过程非常繁琐，例如在应用于 Libras 数据分类时需要构建 $C_{15}^2 = 105$ 个子二元分类器；MKLR 本质上即是一个多分类器，模型构建简易直观，然而实验结果的识别率却并不突出，除在 Waveform 数据集中表现略优外，其余错误率都高于 SVM，究其原因是为了满足大样本应用，训练过程中采用了 Nystrom 近似方法[6]，致使性能有所下降，并且模型不具备稀疏性，所有训练数据都可称之为有效向量；本文所提的 SMKLC 识别性能突出，与 MKLR 一致，也是一种直接多元分类器，并且稀疏性高于 SVM，与 RVM 相近，具有高效的模型运行效率，以 Svmguide2 为例，对于每一个输入样本，SVMOVO 需要 0.015 s 的测试用时，而 SMKLC 模型只需 0.001 s。

除此之外，实验还对模型的构建效率进行了测试，表 3 列出了 SMKLC 与 MKLR 两者在不同条件下的模型训练性能，其中实验 A 条件为：对于同一数据，SMKLC 与 MKLR 达到指定识别率时分别所需的最少训练用时(TM，单位为 s)；实验 B 条件则为：相同数据下，SMKLC 与 MKLR 在指定训练时间下各自的识别错误率。考虑到 RVM、SVM 两者的多元扩展策略较为繁琐，在此不作比较。如表 3 所示，在指定识别精度时，SMKLC 需要更少的模型生成时间，当训练样本数量增长时，优势更为明显；当指定模型训练时间时，SMKLC 的识别性能也优于 MKLR，尤其在 Wine 数据集中，MKLR 原最优错误率为 2.86%，优于 SMKLC 的 6.05%，而当训练时间限制为 2 s 时，MKLR 的错误率提高至 6.8%，劣于 SMKLC 的 6.5%，验证了自下向上训练算法的优越性。

4.3 UCI 基准数据实验 2

SVM 的一个不足是其支持向量数随着训练样本数的增加呈线性增长，为验证 SMKLC 在相同条件下的有效向量增长情况，本文借助 UCI 公共数据集中的 Satimage 和 Waveform 各 4000 例训练样本及 1000 例测试样本，检验 SMKLC 在不同数量训练

表 2 SMKLC 与 SVMOVO, RVMOVO, MKLR 识别率对比

UCI Data	RVMOVO		MKLR		SVMOVO		SMKLC	
	ER(%)	RV	ER(%)	UV	ER(%)	SV	ER(%)	UV
Iris(3 class)	0.00	5.9	0.00	All	0.00	18	0.00	5.5
Wine(3 class)	4.25	12.8	2.86	All	2.25	22.3	6.05	9.5
Libras(15 class)	11.45	15.5	16.32	All	12.75	18.6	11.10	22.7
Svmguide2(3 class)	21.20	42.68	22.58	All	20.20	72.67	19.50	38.5
Waveform(3 class)	9.72	43.2	9.68	All	9.88	90.2	9.59	51

表 3 SMKLC 与 MKLR 模型构建效率对比

UCI Data	实验条件 A			实验条件 B		
	ER(%)	TM(MKLR)(s)	TM(SMKLC)(s)	TM(s)	ER(MKLR)(%)	ER(SMKLC)(%)
Iris	0.0	1.95	1.81	1.5	3.3	1.6
Wine	7.0	2.35	2.12	2.0	6.8	6.5
Libras	17.0	72.0	32.7	50	20.6	11.1
Svmguide2	23.0	1.35	1.10	1.0	25.6	20.1
Waveform	10.0	244	165	150	20.7	14.8

样本情况下的有效向量数变化过程并与 SVM 进行对比。模型训练的结束条件为：在相同训练样本的前提下，两种算法对测试样本的识别率达到一致(差值小于 1%)。实验结果如图 2 所示，其中横坐标为训练样本数，纵坐标为模型训练终止时的支持(有效)向量数。

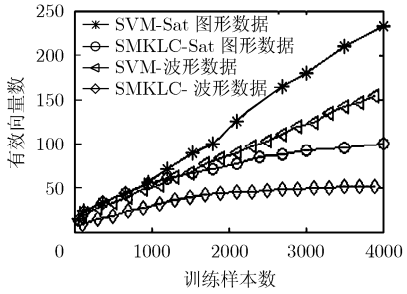

图 2 有效向量数与支持向量数对比

从图 2 可见，在两种不同训练数据下，随着训练样本量的增加，SVM 的支持向量数也近似呈线性增长，当训练样本数量很大时，这一特性将严重影响最终模型的识别实时性能。而 SMKLC 的有效向量数并不随训练样本线性增长，在具体应用中，其有效向量达到一定数值后会呈饱和状态，此后再增加训练样本数将不会影响最终有效向量数。这一特征使 SMKLC 算法在应用于大样本场合时，一旦模型训练完成，即能取得很好的测试实时性。

5 结束语

为弥补经典核 Logistic 回归与支持向量机各自的缺陷，本文提出概率型稀疏核 Logistic 多元分类机，实现了真正的多元分类模型，而非“一对一”或“一对多”等后处理多元化方法，并且在不增加训练负担的前提下为模型参数绑定稀疏性先验概率，强化模型泛化能力。所采用的自下向上训练算法虽然结果仅为次优解，却大大地加速了训练速度，使模型得以应用于大训练样本场合。

实验结果表明，所提的 SMKLC 模型在多分类应用中具有良好的判别效果，其模型稀疏度较 SVM 更高，且有效向量数并不随训练样本的增加而线性增长，对比传统 MKLR 方法，其模型构建速率更快，自下向上进行增补的训练策略也避免了大样本应用时的内存溢出现象。然而，实验过程也表明，SMKLC 在应对不平衡训练数据时效率较低，并且模型中的核函数参数还需经过交叉验证获取，费时较多，未来工作中，将对这两方面进行研究分析，使模型更加完善。

参 考 文 献

[1] 乔立山，陈松灿，王敏. 基于相关向量机的图像阈值技术[J]. 计算机研究与发展, 2010, 47(8): 1329–1337.
Qiao L S, Chen S C, and Wang M. Image thresholding based on relevance vector machine [J]. *Computer Research and Development*, 2010, 47(8): 1329–1337.

[2] Huang F, Jiang Z P, and Zhang S G, *et al*.. Reliability evaluation of wireless sensor networks using logistic regression [C]. ICCMC, Shenzhen, 2010: 334–338.

[3] Li J, Bioucas-Dias J M, and Plaza A. Semi-supervised hyperspectral image segmentation using multinomial logistic regression with active learning [J]. *IEEE Transactions on Geoscience and Remote Sensing*, 2010, 48(11): 4085–4098.

[4] Birkenes O, Matsui T, and Tanabe K, *et al*.. Penalized logistic regression with HMM log-likelihood regressors for speech recognition [J]. *IEEE Transactions on Audio, Speech and Language Processing*, 2010, 18(6): 1440–1454.

[5] Roth V. Probabilistic discriminative kernel classifiers for multi-class problems [C]. Lecture Notes In Computer Science. London, UK, Springer-Verlag, 2001, LNCS 2191: 246–266.

[6] Karsmakers P, Pelckmans K, and Suykens J. Multi-class kernel logistic regression: a fixed-size implementation [C]. Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, 2007: 1756–1761.

[7] Karsmakers P, Pelckmans K, and Suykens J, *et al*.. Fixed-size kernel logistic regression for phoneme classification[C]. Proc. of the Interspeech (INTERSPEECH). 8th Annual Conference of the International Speech Communication Association, Antwerp, Belgium, 2007: 78–81.

[8] Zhu J and Hastie T. Kernel logistic regression and the import vector machine [J]. *Journal of Computational and Graphical Statistics*, 2005, 14(1): 185–205.

[9] Krishnapuram B. Sparse multinomial logistic regression: fast algorithms and generalization bounds [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, 27(6): 957–968.

[10] Fu Zhou-yu and Robles-Kelly A. Fast multiple instance learning via L1, 2 logistic regression [C]. ICPR 2008. The 9th Conference of the International Association for Pattern Recognition, Tampa Convention Center, 2008: 1–4.

[11] Patra S, Shanker K, and Kundu D. Sparse maximum margin logistic regression for credit scoring[C]. ICDM 2008. Eighth IEEE International Conference on Data Mining, Washington DC, USA, 2008: 977–982.

[12] Babacan S D, Molina R, and Katsaggelos A K. Fast Bayesian compressive sensing using laplace priors [C]. ICASSP, Taipei, 2009: 2873–2876.

[13] Tipping M E. Sparse Bayesian learning and the relevance vector machine [J]. *The Journal of Machine Learning Research*, 2001, 1(6): 211–244.

[14] Tzikas D G, Likas A C, and Galatsanos N P. Sparse Bayesian modeling with adaptive kernel learning [J]. *IEEE Transactions on Neural Networks*, 2009, 20(6): 926–937.

[15] Tipping M E and Faul A. Fast marginal likelihood maximization for sparse Bayesian models [C]. Proc. 9th Int. Workshop Artif. Intell. Statist., Key West, FL, 2003: 1–8.

[16] Chamroukhi F, Same A, and Govaert G, *et al*.. A regression model with a hidden logistic process for feature extraction from time series [C]. International Joint Conference on Neural Networks, Atlanta, GA, 2009: 489–496.

[17] MacKay D J C. The evidence framework applied to classification networks [J]. *Neural Computation*, 1992, 4(5): 720–736.

郑建炜： 男，1982 年生，讲师，从事模式识别、人脸跟踪、说话人识别方向的研究.

王万良： 男，1956 年生，教授，研究方向为人工智能、网络控制.

蒋一波： 男，1982 年生，讲师，研究方向为人工智能、网络控制.

陈伟杰： 男，1985 年生，博士生，研究方向为人工智能.