基于稀疏贝叶斯的流形学习

陈兵飞,江兵兵,周熙人,陈欢欢
(中国科学技术大学计算机科学与技术学院,安徽合肥 230027)

摘 要: 针对当前监督学习算法在流形数据集上分类性能的缺陷,如分类精度低且稀疏性有限,本文在稀疏贝叶斯方法和流行正则化框架的基础上,提出一种稀疏流形学习算法(Manifold Learning Based on Sparse Bayesian Approach,MLSBA). 该算法是对稀疏贝叶斯模型的扩展,通过在模型的权值上定义稀疏流形先验,有效利用了样本数据的流形信息,提高了算法的分类准确率. 在多种数据集上进行实验,结果表明:MLSBA 不仅在流形数据集上取得良好的分类性能,而且在非流形数据集上效果也比较好;同时算法在两类数据集上均具有良好的稀疏性能.

关键词: 拉普拉斯; 稀疏贝叶斯; 稀疏流形先验; 流形正则化
中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112 (2018)01-0098-06
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2018.01.014

Manifold Learning Based on Sparse Bayesian Approach

CHEN Bing-fei, JIANG Bing-bing, ZHOU Xi-ren, CHEN Huan-huan
(*School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui* 230027, *China*)

Abstract: Aiming at the classification performance deficiencies of current supervised learning algorithms on manifold data sets, e. g. low classification accuracy and limited sparsity, a sparse manifold learning algorithm based on sparse Bayesian inference and manifold regularization framework is proposed. The algorithm is called manifold learning based on sparse Bayesian approach (MLSBA). MLSBA is an extension of sparse Bayesian model, by introducing sparse manifold priors to the weights, which can effectively employ the manifold information of sample data to improve the classification accuracy. Extensive experiments are conducted on various datasets, and the results show that MLSBA not only achieves better classification performance on manifold datasets, but also has comparable effectiveness on the non-manifold datasets, and our algorithm has good sparsity on two categories of datasets at the same time.

Key words: Laplacian; sparse Bayesian; sparse manifold prior; manifold regularization

1 引言

在监督学习中,对于给定的样本数据 $\boldsymbol{x} \in \mathbf{R}^d$,希望得到一个可以对样本进行分类或者回归的函数 $f(\boldsymbol{x};\boldsymbol{w})$. 为了得到 $f(\boldsymbol{x};\boldsymbol{w})$,首先需要给定一个训练数据集 $\boldsymbol{T}=\{(\boldsymbol{x}_1,\boldsymbol{y}_1),\cdots,(\boldsymbol{x}_N,\boldsymbol{y}_N)\}$(其中输入是一个 d 维向量,$\boldsymbol{x}_i=[x_{i1},\cdots,x_{id}]\in\mathbf{R}^d$,当 $\boldsymbol{y}_i\in\mathbf{R}$ 时,对应的是回归问题;当 $\boldsymbol{y}_i\in\{-1,1\}$ 时,表示研究的是二分类问题);然后在训练数据集上学习得到 $f(\boldsymbol{x};\boldsymbol{w})$;最后在假定与训练数据服从同样分布的测试数据上对 $f(\boldsymbol{x};\boldsymbol{w})$ 的分类性能进行评估.

近年来,基于核方法的机器学习模型引起广泛的关注[1],该模型是权值向量 $\boldsymbol{w}$ 与核函数 $\boldsymbol{\Phi}(\boldsymbol{x})$ 的线性组合[2],如式(1)所示:

$$f(\boldsymbol{x};\boldsymbol{w}) = \sum_{i=1}^{N} \boldsymbol{w}_i\boldsymbol{\varphi}_i(\boldsymbol{x}) + w_0 = \boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{w} \tag{1}$$

其中 w_0 是偏移动,$\boldsymbol{w}=[w_0,w_1,\cdots,w_N]^{\mathrm{T}}$ 是模型的权值向量,$\boldsymbol{\Phi}(\boldsymbol{x})=[\boldsymbol{\varphi}_1(\boldsymbol{x}),\cdots,\boldsymbol{\varphi}_N(\boldsymbol{x})]^{\mathrm{T}}$ 是模型的核函数矩阵. $\boldsymbol{\varphi}_i(x)=[1,k(x_1,x_i),\cdots,k(x_N,x_i)]^{\mathrm{T}}$ 表示由核函数组成的基向量(基函数)[3]. 目前核函数有线性核、高斯核、拉普拉斯核、样条核等多种类型,本文使用高斯核:

$$k(x_i,x_j)=\exp\{-\theta^2\|x_i-x_j\|_2\} \tag{2}$$

其中 $\|\cdot\|_2$ 表示 L_2 范数,θ 是控制核函数的参数,可通过交叉验证给出或最大化后验的方法对其进行优化[1]. 通常情况下,核参数 θ 都会预先给出,此时学习目标由寻找最优的分类函数 $f(\cdot)$ 转变为根据已有的训

收稿日期:2016-04-08;修回日期:2016-09-14;责任编辑:孙瑶
基金项目:国家自然科学基金(No. 91546116, No. 61673363, No. 61511130083)

练数据集 T,估计合适的模型权值参数 $\boldsymbol{w}$. 目前基于核函数的稀疏学习算法包括支持向量机(Support Vector Machines,SVM)[4]、相关向量机(Relevance Vector Machines,RVM)[3]和概率分类向量机(Probabilistic Classification Vector Machines,PCVM)[1]. SVM 通过核函数的映射得到模型的最优解,RVM 和 PCVM 基于稀疏贝叶斯框架,在 $\boldsymbol{w}$ 上引入高斯先验,获得模型的稀疏解和概率输出[1,5].

SVM 使用统计学习理论中的结构风险最小化原则来处理二分类问题. 它将样本数据投影到高维线性可分空间,构造具有较低 VC 维的最优分类超平面作为判别面,使得可分的两类样本的空间间隔最大[6]. 虽然 SVM 被应用在很多领域,且取得良好的效果,但是 SVM 存在着一些明显的不足[1-3,7,8],例如:①由于模型的稀疏性有限,支持向量的个数随训练样本的增长呈线性增长;②预测的结果是无概率输出,无法度量预测结果的不确定性;③需要通过交叉验证选择合适的惩罚因子,增加了模型的计算时间;④核函数必须满足 Mercer 条件[8].

Tipping 等人[3]通过研究 SVM 的缺点,在贝叶斯框架的基础上,给式(1)中 $\boldsymbol{w}$ 的每个分量 w_i 引入相互独立的零均值高斯分布,得到比 SVM 更稀疏且可以产生概率输出的分类模型(RVM). 随后 Chen 等人[1]研究发现,对于类别不同的样本,引入相同的零均值高斯先验会造成模型的不稳定. 文献[1]对不同的类别 $\boldsymbol{y}_i$ 采用不同的截断高斯先验分布,得到了性能更好的贝叶斯分类模型(PCVM). 贝叶斯方法的主要优点是可以有效利用样本数据的先验知识,但是当假设模型与样本数据的实际分布情况不相符时,难以获得较好的性能. 研究发现[7],实际用于学习的一些数据集具有特定的流形结构. 按照流形假设的观点,处于一个很小的局部邻域内的样例具有相似的性质,因此它们的标记也应该相似[9]. 流形假设反映了决策函数的局部平滑性[7]. RVM 是一种稀疏贝叶斯分类器,采用独立的零均值高斯先验,主要解决的是模型的稀疏性问题. 但该模型的缺点是不能有效的利用样本数据所具有的流形信息,因此在流形数据集上难以取得理想的分类效果.

本文基于贝叶斯框架和半监督学习中常用的流形正则化理论,提出一种能够有效利用样本数据流形信息的稀疏流形学习算法(MLSBA). 该方法采用稀疏流形先验,解决了 SVM 中存在的一些问题. MLSBA 集成了传统稀疏贝叶斯分类器的优点,同时能够在流形数据集上取得良好的性能. MLSBA 具有以下优点:①它是基于贝叶斯后验概率的分类器,可以给出概率输出结果,能够度量预测结果的不确定性;②新的学习算法同样是稀疏的,因此模型在估计权值向量时具有很好的稀疏性能,这种稀疏性降低了测试阶段的计算复杂度;③通过定义新的先验分布,将样本数据所具有的流形信息作为先验知识引入到模型中,能够对分类函数进行约束,从而提升了算法在流形数据集上的分类性能.

2　稀疏贝叶斯流形学习

2.1　模型描述

在二分类问题中,输入的数据集为 $\boldsymbol{T}=\{(\boldsymbol{x}_1,y_1),\cdots,(\boldsymbol{x}_N,y_N)\}$,其中标记 $y_i\in\{-1,1\}$. 对 $\boldsymbol{T}$ 进行分类时,模型需要使用一个连接函数,该函数可以将输出的结果映射到二值变量 $\{-1,1\}$ 中,本文参考 RVM 分类模型,采用 Sigmoid 激活函数,表达式为:

$$\psi(z)=\frac{1}{1+\exp(-z)} \tag{3}$$

由于使用 Sigmoid 函数,下一节的拉普拉斯近似推导将变得更容易[10,11]. 通过激活函数和式(1)的组合,MLSBA 分类模型为:

$$p(\boldsymbol{y}=1\mid\boldsymbol{x},\boldsymbol{w})=\psi(\boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{w}) \tag{4}$$

其中 $\boldsymbol{\Phi}(\boldsymbol{x})$ 不需要满足 Mercer 条件[3].

连接函数确定后,需要确定模型的似然函数[1],分类模型中经常使用的似然函数有贝努利似然和高斯似然,本文选择贝努利似然函数:

$$p(\boldsymbol{t}\mid\boldsymbol{w})=\prod_{i=1}^{N}\psi_i^{t_i}[1-\psi_i]^{1-t_i} \tag{5}$$

其中 $t_i=(y_i+1)/2\in\{0,1\}$,$\boldsymbol{t}=[t_1,\cdots,t_N]^{\mathrm{T}}$ 表示目标值,$\psi_i=\psi(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}_i(\boldsymbol{x}))$.

2.2　稀疏流形先验

RVM 为了获得稀疏解,给每一个权值分量 w_i 分别定义一个零均值高斯先验[12]. 但是,RVM 只考虑模型的稀疏性,它认为模型总是越稀疏越好,忽略了样本数据本身具有的流形结构对模型的影响. 为了让模型既具有较好的稀疏性能又能有效的利用样本数据的流形结构信息,本文基于 RVM 算法与流形正则化理论,在权值向量 $\boldsymbol{w}$ 上定义稀疏流形先验:

$$p(\boldsymbol{w}\mid\boldsymbol{\alpha},\lambda)=(2\pi)^{-(N+1)/2}|\boldsymbol{A}+\lambda\boldsymbol{B}|^{1/2}*\exp\left\{-\frac{1}{2}\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{A}+\lambda\boldsymbol{B})\boldsymbol{w}\right\} \tag{6}$$

其中 $\boldsymbol{\alpha}=[\alpha_0,\alpha_1,\cdots,\alpha_N]^{\mathrm{T}}$ 表示控制权值向量 $\boldsymbol{w}$ 的超参数向量,它服从 Gamma 分布[13]. 矩阵 $\boldsymbol{A}$ 是由 α_i 组成的对角矩阵,$\boldsymbol{A}=\mathrm{diag}\{\alpha_0,\alpha_1,\cdots,\alpha_N\}$. λ 是控制模型利用样本数据流形信息的参数,它也服从 Gamma 分布,当 $\lambda\propto 0$ 时,表示模型没有使用样本数据的流形结构信息,算法变成标准的 RVM 分类算法. $\boldsymbol{B}=\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{\Phi}$($\boldsymbol{\Phi}$ 为核函数矩阵 $\boldsymbol{\Phi}(\boldsymbol{x})$ 的简写),它与 $\boldsymbol{w}$ 构成流形正则项. 因此引入 $\boldsymbol{B}$ 可以将样本数据的流形信息作为先验知识融入到模型中. 式(6)中 $\boldsymbol{L}=\boldsymbol{C}-\boldsymbol{S}$ 叫做图的拉普拉斯矩阵[14],

通过它可以获取样本数据所具有的流形结构. $\boldsymbol{S}$ 是近邻矩阵,它的元素 S_{ij} 度量样本 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 之间的相似性. $\boldsymbol{C}$ 为对角矩阵,其中 $C_{ii}=\sum_j S_{ij}$. 为高效计算图拉普拉斯矩阵,本文构建一个 kNN 图[7],其中近邻矩阵计算方式如下:

$$S_{ij}=\begin{cases}\exp\left\{-\dfrac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{\zeta^2}\right\}, & \text{if } \boldsymbol{x}_i\in \text{kNN}(\boldsymbol{x}_j)\\ 0, & \text{if otherwise}\end{cases} \tag{7}$$

其中 $\text{kNN}(\boldsymbol{x}_j)$ 表示距离样本 $\boldsymbol{x}_j$ 最近的 k 个样本集,ζ 为宽度参数,一般取 k 近邻样本之间的距离均值[7]. 可以看出,近邻矩阵 $\boldsymbol{S}$ 是一个稀疏矩阵,因此图拉普拉斯矩阵 $\boldsymbol{L}$ 也是稀疏的.

2.3 拉普拉斯近似

根据贝叶斯理论,当 $\boldsymbol{\alpha},\lambda$ 一定时,通过式(6)中的先验和式(5)中的似然函数,可以计算出权值向量的后验概率分布,如式(8)所示:

$$p(\boldsymbol{w}|\boldsymbol{t})=\frac{p(\boldsymbol{t}|\boldsymbol{w})p(\boldsymbol{w}|\boldsymbol{\alpha},\lambda)}{p(\boldsymbol{t}|\boldsymbol{\alpha},\lambda)} \tag{8}$$

在式(8)中,由于似然函数 $p(\boldsymbol{t}|\boldsymbol{w})$ 不服从高斯分布,因此后验分布 $p(\boldsymbol{w}|\boldsymbol{t})$ 无法直接求出. 为了得到 $\boldsymbol{w}$ 的最大后验概率分布,需要使用拉普拉斯近似算法[8]. 拉普拉斯近似是一种确定的近似算法,它用一个高斯分布去近似已知的概率分布.

根据式(8),$\boldsymbol{w}$ 的最大后验估计记为 $\boldsymbol{w}^*$(即拉普拉斯近似中 $\boldsymbol{w}$ 的众数位置 $\boldsymbol{\mu}$). 可对后验 $p(\boldsymbol{w}|\boldsymbol{t})$ 取自然对数,有:

$$\begin{aligned}\boldsymbol{Q}&=\ln\{p(\boldsymbol{t}|\boldsymbol{w})p(\boldsymbol{w}|\boldsymbol{\alpha},\lambda)\}-\ln\{p(\boldsymbol{t}|\boldsymbol{\alpha},\lambda)\}\\&=\sum_{i=1}^{N}[t_i\ln\psi_i+(1-t_i)\ln(1-\psi_i)]\\&\quad-\frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w}-\frac{\lambda}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{w}+\text{const}\end{aligned}$$

其中 $\boldsymbol{Q}$ 为带正则项的对数似然函数,可以通过迭代重复加权最小二乘法(IRLS)求解[3,4]. 计算 $\boldsymbol{Q}$ 关于 $\boldsymbol{w}$ 的梯度一阶偏导数和二阶偏导数,分别为:

$$\frac{\partial\boldsymbol{Q}}{\partial\boldsymbol{w}}=\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{t}-\boldsymbol{\psi})-(\boldsymbol{A}+\lambda\boldsymbol{B})\boldsymbol{w}$$

$$\frac{\partial^2\boldsymbol{Q}}{\partial\boldsymbol{w}^2}=-(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{\Phi}+\boldsymbol{A}+\lambda\boldsymbol{B})$$

其中 $\boldsymbol{\psi}=[\psi_1,\cdots,\psi_N]^{\mathrm{T}}$,$D=\text{diag}\{d_1,\cdots,d_N\}$,$d_i=\psi_i(1-\psi_i)$.

拉普拉斯近似用一个高斯分布对后验概率分布在众数位置 $\boldsymbol{\mu}$ 处的函数进行二次逼近. 当 IRLS 算法收敛时,得到以 $\boldsymbol{\mu}$ 为中心的高斯分布,其均值、方差分别为:

$$\boldsymbol{w}^*=(\boldsymbol{A}+\lambda\boldsymbol{B})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{t}-\boldsymbol{\psi}) \tag{9}$$

$$\boldsymbol{\Sigma}=(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{\Phi}+\boldsymbol{A}+\lambda\boldsymbol{B})^{-1} \tag{10}$$

通过在均值和方差中引入表示样本数据流形信息的正则项 $\boldsymbol{B}$ 以及控制模型利用样本数据流形的参数 λ,能够充分利用样本数据信息.

2.4 超参数优化

与 RVM 类似,MLSBA 学习最终归结为求解超参数 $\boldsymbol{\alpha},\lambda$ 的后验概率分布 $p(\boldsymbol{\alpha},\lambda|\boldsymbol{t})$ 的最大值[1]. 由于令 $p(\boldsymbol{\alpha})$、$p(\lambda)$ 为无信息 Gamma 分布[11],则有 $p(\boldsymbol{\alpha},\lambda|\boldsymbol{t})\propto p(\boldsymbol{t}|\boldsymbol{\alpha},\lambda)p(\boldsymbol{\alpha})p(\lambda)$[3,10],因此求解 $p(\boldsymbol{\alpha},\lambda|\boldsymbol{t})$ 的最大值转化为最大化边际似然函数 $p(\boldsymbol{t}|\boldsymbol{\alpha},\lambda)$. 根据上一节的拉普拉斯近似结果,边际似然函数为:

$$\begin{aligned}p(\boldsymbol{t}|\boldsymbol{\alpha},\lambda)&=\int p(\boldsymbol{t}|\boldsymbol{w})p(\boldsymbol{w}|\boldsymbol{\alpha},\lambda)\,\mathrm{d}\,\boldsymbol{w}\\&\approx p(\boldsymbol{t}|\boldsymbol{w}^*)p(\boldsymbol{w}^*|\boldsymbol{\alpha},\lambda)(2\pi)^{N/2}|\boldsymbol{\Sigma}|^{1/2}\end{aligned}$$

于是有:

$$\begin{aligned}\boldsymbol{Z}&=\ln p(\boldsymbol{t}|\boldsymbol{\alpha},\lambda)=\frac{1}{2}\ln|\boldsymbol{A}+\lambda\boldsymbol{B}|+\frac{1}{2}\ln|\boldsymbol{\Sigma}|\\&\quad-\frac{1}{2}(\boldsymbol{w}^*)^{\mathrm{T}}(\boldsymbol{A}+\lambda\boldsymbol{B})\boldsymbol{w}^*\end{aligned}$$

令 $\dfrac{\partial\boldsymbol{Z}}{\partial\ln\boldsymbol{\alpha}_i}=0$,$\dfrac{\partial\boldsymbol{Z}}{\partial\ln\lambda}=0$,得到:

$$\begin{aligned}\alpha_i^{\text{new}}&=\frac{\gamma_i}{w_i^{*}+M_{ii}}\\\lambda^{\text{new}}&=\frac{\text{trace}[\boldsymbol{I}-(\boldsymbol{A}+\lambda\boldsymbol{B})^{-1}\boldsymbol{A}]}{\text{trace}(\boldsymbol{\Sigma}\boldsymbol{B})+(\boldsymbol{w}^*)^{\mathrm{T}}\boldsymbol{B}\boldsymbol{w}}\end{aligned} \tag{11}$$

其中 $\boldsymbol{M}=\lambda\boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{I}+\lambda\boldsymbol{A}^{-1}\boldsymbol{B})^{-1}$,$M_{ii}$ 表示矩阵 $\boldsymbol{M}$ 的第 i 个对角元素,$\boldsymbol{\alpha}$ 控制模型的稀疏性,λ 控制模型对样本数据流形信息的利用程度. 通过最大化边际似然函数对超参数 $\boldsymbol{\alpha}$ 和 λ 进行优化,既实现了模型的稀疏性,又可以让模型根据样本数据本身的分布情况自动选择超参数 λ. 因此模型在不具有明显流形结构的数据集上,也能取得较好的分类结果.

MLSBA 的学习过程,就是对式(11)进行迭代,给出 α_i^{new} 和 λ^{new},然后用 α_i^{new}、λ^{new} 对式(9)和式(10)中的 $\boldsymbol{w}^*$ 和 $\boldsymbol{\Sigma}$ 进行更新,直到满足收敛条件. 在实际迭代更新过程中,许多超参数 α_i 会变的很大,此时 α_i 控制的权值 w_i 趋于零,为实现模型的稀疏性,以降低模型的时间复杂度,可将 w_i 置为零,同时将它对应的基向量 $\boldsymbol{\varphi}_i$ 从模型中修剪出去.

3 实验结果与分析

为了验证 MLSBA 的有效性,本文将在不同的数据集上验证算法的性能,并将其与典型的监督分类算法进行对比. 首先将 MLSBA 在二维合成数据集上与 RVM 和 PCVM 进行比较;然后将 MLSBA 应用在 8 个 benchmark 数据集上;最后我们选择 5 个具有流形结构的数据集对算法的性能进行测试. 本文实验的数据集信息见表 1 所示. 其中,USPS 是一个十类数据集,将前五类作为正类样本,后五类作为负类样本,由此得到它的 2

分类形式 $USPS_2$. 实验使用分类错误率作为检验算法性能的标准.

表 1　实验数据集描述

数据集	训练样本	测试样本	正类%	负类%	维度
Synth	250	1000	50.00%	50.00%	2
two-moon	50	350	52.75%	47.25%	2
Banana	400	4900	44.83%	55.17%	2
Cancer	200	77	29.28%	70.72%	9
Diabetics	468	300	34.90%	65.10%	8
Ringnorm	400	7000	49.51%	50.49%	20
German	700	300	30.00%	70.00%	20
Heart	170	100	44.44%	55.56%	13
Titanic	150	2051	58.33%	41.67%	3
Twonorm	400	7000	50.04%	49.96%	20
G50c	50	500	50.00%	50.00%	50
G10n	700	300	49.82%	50.18%	10
Mnist3vs8	500	13466	48.87%	51.13%	784
PCMAC	100	1843	50.54%	49.46%	3289
$USPS_2$	400	7000	41.63%	58.37%	256

3.1　二维合成数据分类测试

本节采用两个二维合成数据集来验证 MLSBA 的学习性能. 数据集分别为 Synth 数据集和 two-moon 数据集(具体描述见表 1). 其中 Synth 数据集不具有明显的流形结构,它由两个部分重合的二维高斯分布混合得到,有着 8% 的固有错误率;two-moon 数据集具有明显的流形结构. 实验结果如图 1 表示,从图 1 可以看出,MLSBA 在 Synth 数据集上能够取得与 RVM、PCVM 相当的分类性能,但在 two-moon 数据集上 MLSBA 的性能明显优于另外两个算法,这充分验证了 MLSBA 在流形数据集上更有效.

3.2　Benchmark 数据分类测试

本节实验主要是测试 MLSBA 在 Benchmark 数据集上的分类性能. 实验选择 8 个数据集(详情见表 1),其中实验数据集已进行预处理. 实验中用到的对比算法,

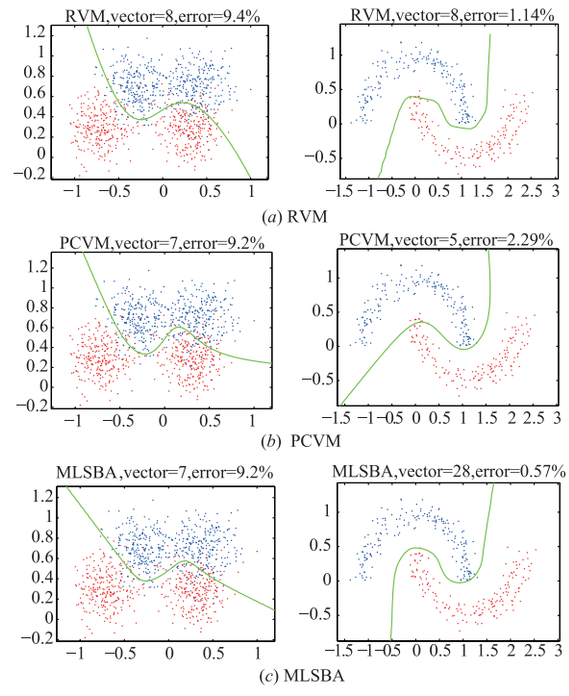

图1　RVM、PCVM、MLSBA在二维数据集上的分类结果,左边是Synth数据集,右边是two-moon数据集

如 kNN,它的参数 k 从 $\{1,2,\cdots,20\}$ 中选择;ELM[15] 的隐层节点个数从 $\{50,100,\cdots,500\}$ 中选择;LDA 按照文献[1]中的方式设置参数. 使用核函数的算法(RVM,SVM,MLSBA,PCVM)统一采用标准的高斯核函数,根据式(2)定义,其中核参数通过二折交叉验证法确定. 各算法分别在 Benchmark 数据集上运行 100 次,平均分类错误率和标准差如表 2 所示.

表 2 中的实验结果表明:MLSBA 在不具有流形结构的数据集上同样能取得较好的分类效果. MLSBA 使用超参数 λ 控制模型对数据流形信息的利用程度,其中 Benchmark 数据集不具有明显的类内流形结构,因此算法在迭代的过程中,λ 会逐渐更新收敛到零,此时算法的性能与 RVM 相当.

表 2　Benchmark 数据集上所有算法的平均分类错误率(%)和标准差

算法	Banana	Cancer	Diabetics	Ringnorm	German	Heart	Titanic	Twonorm
1NN	13.64(0.76)	32.70(4.84)	30.12(2.05)	35.03(1.36)	29.46(2.47)	23.16(3.74)	33.00(11.92)	6.68(0.72)
kNN	11.26(0.54)	30.58(4.19)	25.83(1.90)	35.03(1.36)	25.44(2.52)	**16.46**(3.61)	23.84(4.94)	2.81(0.20)
LDA	46.69(4.81)	31.81(4.35)	24.66(2.03)	24.62(0.66)	28.52(2.56)	16.60(2.87)	23.57(4.61)	2.61(0.17)
ELM	11.03(0.59)	28.01(4.82)	24.31(1.80)	22.49(1.14)	**23.19**(2.22)	17.84(3.39)	22.33(1.01)	3.72(0.34)
SVM	11.98(0.71)	28.97(4.87)	30.68(2.28)	1.66(0.12)	26.63(2.35)	22.29(3.50)	**22.30**(1.05)	2.93(0.27)
RVM	10.78(0.52)	26.60(4.70)	23.81(1.84)	2.15(0.64)	24.52(2.31)	17.30(3.56)	23.30(1.50)	3.32(0.43)
PCVM	**10.30**(0.76)	26.23(4.62)	23.68(2.08)	**1.53**(0.13)	23.62(2.24)	16.62(3.45)	22.58(1.37)	**2.46**(0.26)
MLSBA	10.65(0.46)	**26.13**(4.81)	**23.24**(1.82)	2.11(0.29)	23.83(2.33)	17.04(2.96)	22.75(1.14)	3.48(0.47)

3.3 流形数据类测试

本节将选择 5 个具有流形结构的数据集,对 MLSBA 进行测试,验证 MLSBA 在流形数据集上的有效性. 实验数据集的详细描述见表 1. 实验算法的参数设置与上节实验相同,各算法分别在流形数据集上运行 100 次,实验结果如表 3 所示.

从表 3 中可以看出,MLSBA 在 5 个实验数据集中的 4 个上取得最低的测试错误率,在 $USPS_2$ 上取得了较好分类结果. 实验结果表明,MLSBA 算法通过构建图拉普拉斯矩阵获取样本类内流形信息,并将其作为模型的先验知识,从而有效降低算法的分类错误率. 相比目前主流的监督学习算法(SVM、RVM、PCVM),MLSBA 在流形数据集上的分类性能有较大提升.

表 3 流形数据集上所有算法的平均分类错误率(%)

算法	G50c	G10n	Mnist3vs8	PCMAC	$USPS_2$
1NN	23.50	29.79	8.31	39.83	6.37
kNN	14.26	25.60	7.53	38.75	6.37
ELM	22.55	19.59	10.57	30.84	16.44
SVM	14.15	22.58	4.28	27.69	**5.77**
RVM	15.65	12.88	5.21	28.25	8.73
PCVM	17.83	17.72	5.84	34.61	12.70
MLSBA	**13.76**	**12.17**	**4.04**	**24.96**	6.69

3.4 模型的稀疏性

稀疏模型能够有效的提取数据集中重要样本,简化模型,提高算法的泛化性能. 本节将给出以上两节实验中用到的平均样本数,以此度量模型的稀疏性,实验结果如表 4 所示. 从表 4 中可以看出,SVM 用到的支持向量个数随训练样本数呈线性增长,而 RVM 和 PCVM 使用相对较少的关联向量. MLSBA 在大部分实验数据集上的稀疏性与 RVM 和 PCVM 相当,虽然在 Titanic、Mnist3vs8、$USPS_2$ 上的稀疏性较差,但是仍然优于 SVM.

表 4 算法利用的平均样本数和标准差

数据集	训练样本	SVM	RVM	PCVM	MLSBA
Banana	400	88.3 ±10.4	12.7 ±1.5	19.8 ±4.3	**5.1** ±1.5
Cancer	200	105.7 ±6.4	9.4 ±2.0	9.4 ±2.6	**4.7** ±1.0
Diabetics	468	376.0 ±7.0	8.4 ±2.0	19.8 ±3.1	**4.7** ±0.8
Ringnorm	400	151.2 ±8.9	6.5 ±1.7	17.6 ±3.8	**5.9** ±1.4
German	700	514.1 ±12.4	27.4 ±4.3	40.9 ±7.9	**9.1** ±1.2
Heart	170	104.8 ±6.1	9.3 ±1.8	5.9 ±2.2	**5.2** ±1.0
Titanic	150	143.5 ±6.3	**6.8** ±1.4	16.6 ±2.4	38.2 ±9.6
Twonorm	400	236.9 ±8.4	7.7 ±1.1	13.8 ±3.4	**7.5** ±1.1
G50c	50	46.8 ±3.1	6.3 ±1.2	**5.5** ±1.0	6.8 ±0.8
G10n	50	46.1 ±4.2	**5.3** ±1.0	6.4 ±0.5	6.2 ±0.8
Mnist3vs8	500	407.4 ±7.6	17.5 ±3.0	**15.5** ±3.5	189.4 ±5.7
PCMAC	100	99.1 ±1.1	14.4 ±4.8	**8.7** ±1.4	10.8 ±2.4
$USPS_2$	400	383.1 ±4.8	**23.6** ±3.0	32.3 ±4.3	55.4 ±7.1

4 算法分析

本文提出的算法时间复杂度为 $O(N^3)$,其中 N 为训练数据的个数. 由于计算权值参数 $\boldsymbol{w}$ 需要对矩阵求逆,为了避免直接求逆出现数值不稳定,本文算法使用 Cholesky 分解[1,3],它的计算复杂度为 $O(m^3)$,存储复杂度为 $O(m^2)$,其中 m 为模型中基函数的个数,初始时 $m=N$. 由于模型的先验具有良好的稀疏性,因此在大多数情况下,MLSBA 可以将基函数的数量快速地从初始时的 N 修剪到很小的规模. 另外,与 SVM 相比,MLSBA 不需要选择惩罚因子,有效地减少了计算时间. 在 RVM 算法中,通过最大化权值向量 $\boldsymbol{w}$ 的后验概率得到的权值参数 $\boldsymbol{w}$ 的更新公式 $\boldsymbol{w}_{\text{rvm}}=\boldsymbol{A}^{-1}\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{t}-\boldsymbol{y})$. 在 MLSBA 中,$\boldsymbol{w}_{\text{mlsba}}=(\boldsymbol{A}+\lambda\boldsymbol{B})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{t}-\boldsymbol{y})$. 因为 $\lambda\geqslant 0$,且 $\boldsymbol{B}$ 是半正定的,所以 $|\boldsymbol{A}^{-1}|\leqslant|(\boldsymbol{A}+\lambda\boldsymbol{B})^{-1}|$. 因此在同样的条件下,按照 $\boldsymbol{w}_{\text{mlsba}}$ 更新 $\boldsymbol{w}$,将使那些与模型无关的基函数对应的权值 $\boldsymbol{w}_i$ 会以更快的速度收敛到零. 因此 MLSBA 的收敛速度比 RVM 更快.

表 5 给出了 SVM、RVM、PCVM、MLSBA 在 5 个流形数据集上运行 100 次后的平均运行时间. 实验结果表明,相比算法 RVM、PCVM,MLSBA 的运行时间更短,收敛速度更快,与上文的算法计算时间分析一致. 表中 SVM 的运行速度比 MLSBA 快,是因为 SVM 运行时使用了 MATLAB 工具箱和 C++ MEX 文件[16]. 综上可以看出,MLSBA 算法不仅具有较低的分类错误率,而且还具有较快的运行速度.

表 5 算法 SVM、RVM、PCVM、MLSBA 在流形数据集上的平均运行时间(s)

算法	G50c	G10n	Mnist3vs8	PCMAC	$USPS_2$
SVM	0.08	0.08	3.01	0.29	0.86
RVM	0.13	0.11	6.32	0.86	1.12
PCVM	0.69	0.68	8.87	2.67	5.32
MLSBA	0.10	0.08	3.54	0.47	1.05

5 结论

本文基于贝叶斯框架与流形正则化理论,提出了一种新的稀疏分类算法 MLSBA. 该算法使用稀疏贝叶斯推理模型,并结合流形正则化框架中对流形正则项的定义,在模型的参数上定义稀疏流形先验. 实验结果表明,本文所提出的算法具有良好的稀疏性能,能够有效的利用样本数据的流形信息提升算法的分类性能,同时对于不具有类内流形结构的数据,也能取得较好的分类效果. 最后对算法的复杂度和收敛速度进行分析,同时给出了算法运行时间的结果,表明本文算法复杂度与 RVM 相当,但是收敛速度比 RVM 更快.

本算法具有开放性,在未来的研究中可将它进一步扩展到回归和特征选择. 此外,由于算法仍具有较高的计算复杂度,希望在后续的研究中对其进行优化,降低算法的计算复杂度,实现在大规模数据集上的应用.

参考文献

[1] CHEN Huanhuan, TIŇO Peter, YAO Xin. Probabilistic classification vector machines[J]. IEEE Transactions on Neural Networks,2009,20(6):901 – 914.

[2] BISHOP C M. Pattern Recognition and Machine Learning [M]. New York, SA: Springer Science & Business Media,2013.

[3] TIPPING Michael E. Sparse Bayesian learning and the relevance vector machine[J]. Journal of Machine Learning Research,2001,1:211 – 244.

[4] BERGER J O. Statistical Decision Theory and Bayesian Analysis[M]. New York,USA:Springer Science & Business Media,2013.

[5] LI Chang,CHEN Huanhuan. Sparse Bayesian approach for feature selection[A]. IEEE Symposium on Computational Intelligence in Big Data[C]. USA: IEEE Press, 2014. 1 – 7.

[6] PLATT J. Fast training of support vector machines using sequential minimal optimization[A]. Advances in Kernel Methods Support Vector Learning[C]. USA: MIT Press, 1999. 185 – 208.

[7] CHEN Lin, TSANG I W, XU D. Laplacian embedded regression for scalable manifold regularization [J]. IEEE Transactions on Neural Networks and Learning Systems, 2012,23(6):902 – 915.

[8] 杨国鹏,周欣,余旭初,陈伟. 基于相关向量机的高光谱影混合像元分解[J]. 电子学报,2010,38(12):2751 – 2756.
YANG Guopeng,ZHOU Xin,YU Xuchu,CHEN Wei. Relevance vector machine for hyperspectral imagery unmixing [J]. Acta Electronica Sinica,2010,38(12):2751 – 2756. (in Chinese)

[9] MIKHAIL Belkin, PARTHA Niyogi, VIKAS Sindhwani. Manifold regularization:A geometric framework for learning from labeled and unlabeled examples[J]. Journal of Machine Learning Research,2006,7:2399 – 2434.

[10] TZIKAS D G,LIKAS C L,GALATSANOS N P. Sparse Bayesian modeling with adaptive kernel learning [J]. IEEE Transactions on Neural Networks,2009,20(6):926 – 937.

[11] 成萍,司锡才,姜义成,徐荣庆. 基于稀疏贝叶斯学习的稀疏信号表示 ISAR 成像方法[J]. 电子学报,2008,36(3):547 – 550.
CHENG Ping, SI Xicai, JIANG Yicheng, XU Rongqing. Sparse signal representation ISAR imaging method based on sparse Bayesian learning[J]. Acta Electronica Sinica, 2008,36(3):547 – 550. (in Chinese)

[12] TIPPING Michael E,FAUL A . Analysis of sparse Bayesian learning[A]. Proceedings of the Conference on Neural Information Processing Systems[C]. Vancouver, British Columbia,Canada,2002. 383 – 389.

[13] 王天云,陆新飞,丁丽,尹治平,陈卫东. 基于贝叶斯压缩感知的 FD-MIMO 雷达 Off-Grid 目标稀疏成像[J]. 电子学报,2016,44(6):1314 – 1321.
WANG Tianyun, LU Xinfei, DING Li, YIN Zhiping, CHEN Weidong. Bayesian compressive sensing-based sparse imaging for off-grid target in frequency diverse MIMO radar[J]. Acta Electronica Sinica,2016,44(6): 1314 – 1321. (in Chinese)

[14] ZHU Xiaojin, LAFFERTY John, ROSENFELD Ronald. Semi-Supervised Learning with Graphs[M]. Diss Carnegie Mellon University, Language Technologies Institute, School of Computer Science,2005.

[15] HUANG Guangbin,ZHU Q Y,SIEW C K. Extreme learning machine: theory and applications [J]. Neurocomputing,2006,70(1):489 – 501.

[16] CHEN Huanhuan,TIŇO Peter,YAO Xin. Efficient probabilistic classification vector machine with incremental basis function selection[J] . IEEE Transactions on Neural Networks and Learning Systems,2014,25(2):356 – 369.

作者简介

陈兵飞　男,1991 年出生于安徽池州,中国科学技术大学计算机科学与技术学院在读硕士研究生,研究方向为机器学习.
E-mail:cbf1@mail.ustc.edu.cn

江兵兵(通信作者)　男,1991 年出生于安徽阜阳,现为中国科学技术大学计算机科学与技术学院在读博士研究生,研究方向为贝叶斯学习、半监督学习、特征选择等.
E-mail:jiangbb@mail.ustc.edu.cn