文章编号: 1001-0920 (2014) 09-1587-06 **DOI:** 10.13195/j.kzyjc.2013.0313

一种新高斯过程分类算法

贺建军[1,2], 张俊星[1], 贾思齐[3], 刘文鹏[1], 许 爽[1], 崔艳秋[1]

(1. 大连民族学院 信息与通信工程学院，辽宁 大连 116600；2. 大连理工大学 电子信息与电气工程学部，辽宁 大连 116024；3. 立命馆大学 情报理工学部，草津 520-2102)

摘 要: 由于需要利用高斯函数逼近潜变量函数的后验概率, 传统高斯过程分类算法通常都存在计算复杂度高的问题. 对此, 提出一种新高斯过程分类算法. 该算法的基本思想为: 首先, 利用 Parzen 窗方法估计出每个训练样本的后验概率; 然后, 通过所得到的后验概率将原始分类问题变换为回归问题; 进而分析地得到潜变量函数后验概率的显式表达式, 以避免逼近后验概率所面临的高计算复杂度问题. 仿真实验结果表明, 所提出的算法在分类精度上优于已有的高斯过程分类算法.

关键词: 高斯过程模型；二分类；后验概率；贝叶斯方法

中图分类号: TP273 **文献标志码:** A

A new Gaussian process classification algorithm

HE Jian-jun[1,2], *ZHANG Jun-xing*[1], *JIA Si-qi*[3], *LIU Wen-peng*[1], *XU Shuang*[1], *CUI Yan-qiu*[1]

(1. College of Information and Communication Engineering，Dalian Nationalities University，Dalian 116600，China; 2. Faculty of Electronic Information and Electrical Engineering，Dalian University of Technology，Dalian 116024，China；3. Graduate School of Information Science and Engineering，Ritsumeikan University，Kusatsu 520-2102，Japan．Correspondent：ZHANG Jun-xing，E-mail：zhangjunxing@dlnu.edu.cn)

Abstract: Because the posterior probability of the latent function needs to be approximated by a tractable Gaussian function, the traditional Gaussian process classification algorithms usually suffer from high computational cost. Therefore, a new Gaussian process classification algorithm is proposed. The basic idea is to use Parzen-window method to estimate the posterior probability of training data, and then transform the classification problem to a regression problem based on the obtained posterior probability. As a result, the explicit expression of the posterior probability of the latent function can be derived analytically and the high computational cost caused by approximating the posterior probability with Gaussian distribution is also avoided. The experimental results show that the proposed algorithm can achieve superior classification accuracy to the existing Gaussian process classification algorithms.

Key words: Gaussian process model；binary classification；posterior probability；Bayesian approach

0 引 言

由于基于核函数的机器学习方法 (核方法)[1-3]在处理高维、非线性问题时具有其他方法 (如 K-近邻、人工神经网络和决策树等) 无法比拟的优越性, 在近 20 年受到了各个领域学者的广泛关注. 高斯过程模型[4-7]是近几年在对贝叶斯人工神经网络的研究过程中逐渐发展起来的一种新的核方法, 除传统核方法的各种优点外, 它还具有完全的贝叶斯公式化表示、易实现和可自适应地获得参数等优点. 鉴于高斯过程模型的诸多优点, 现已被广泛地用于构建回归[8]、分类[9-11]、关系学习[12]、强化学习[13]、半监督学习[14]、排序学习[15]、多任务学习[16], 以及多示例多标签学习[17-18]等各种机器学习问题的学习算法.

高斯过程模型主要包括定义潜变量函数、定义似然函数和计算潜变量函数的后验概率 3 个主要模块[19]. 在回归问题中, 所定义的似然函数通常是一个高斯型函数, 所以可以分析地得到潜变量函数后验概率的表达式. 然而, 在分类问题中, 由于所

收稿日期: 2013-03-24；**修回日期:** 2013-07-26.

基金项目: 国家自然科学基金项目 (61374170)；国家民委科研项目 (12DLZ018, 12DLZ001, 2013-GM-003)；辽宁省教育厅科学技术研究项目 (L2012479, L2013504)；中央高校基本科研业务费专项资金项目 (DC13010216, DC120101131).

作者简介: 贺建军 (1983−), 男, 讲师, 博士, 从事机器学习、语音信号处理等研究；张俊星 (1969−), 男, 教授, 从事民族信息化、机器学习等研究.

定义的似然函数通常是一个S形函数, 并不能分析地得到潜变量函数后验概率的表达式. 已有的高斯过程分类算法主要通过以下两类途径解决该问题. 一类的基本思想是利用一个高斯函数来逼近潜变量函数的后验概率, 由于采用不同的逼近算法所得到的分类算法的性能会不同, 人们先后提出了许多方法, 例如: LA (Laplace) 逼近方法[9]、EP (expectation propagation) 逼近方法[10]、KL (Kullback-Leibler) 散度最小化方法[11]、VB (variational bounding) 方法[20]、mean field 方法[21], 以及 Markov chain Monte Carlo 方法[22]等. 虽然这些逼近方法在有的问题中能取得良好的分类效果, 但它们都存在计算复杂度高的问题. 就LA、VB 和EP 方法而言, 在解决包含 n 个训练样本的问题时, 训练模型的计算复杂度是 $O(ln^3)$, 其中 l 表示模型求解算法的迭代次数. 另一类的基本思想是想办法定义高斯型的似然函数, 从而可以得到后验概率的分析表达式. 文献 [23] 提出了一种 Twin 高斯过程分类算法, 该算法的基本思想是为每一类样本定义一个潜变量函数, 要求这些潜变量函数在其所对应类的样本上的值尽可能地小, 而在其他类的样本上的值尽可能地大, 这样即可定义一种高斯型似然函数, 从而得到潜变量函数后验概率的分析表达式. 虽然, Twin 高斯过程分类算法通过设计一种不同结构的潜变量函数, 巧妙地在不借助任何逼近措施的情况下推导出后验概率的分析表达式, 从而避免逼近后验概率所带来的高计算复杂度问题, 但却带来了另外一个问题, 即预测函数的计算复杂度太高. 虽然本文给出了一种近似计算方法, 但正如在结论中所指出的, 该方法缺乏理论上的支持.

本文受人们最近提出的后验概率支持向量机[24-25]的启发, 提出一种新的高斯过程分类算法, 称其为后验概率高斯过程分类算法. 该算法的基本思想是: 首先计算得到每个训练样本的后验概率, 通过所得到的后验概率将原始分类问题变换为回归问题; 然后, 利用高斯过程回归算法分析地得到潜变量函数后验概率的表达式, 进而直接得到预测函数的表达式. 与前面提到的第 1 类高斯过程分类算法相比, 由于本文算法可直接得到潜变量函数后验概率的分析表达式, 避免了逼近后验概率所遭遇的高计算复杂度的问题; 与前面第 2 类高斯过程分类算法相比, 由于所得到的预测函数的计算复杂度本身就很小, 避免了预测函数计算复杂度高的问题. 另外, 虽然本文所建立的后验概率高斯过程分类算法在一定程度上受到了文献 [24-25] 的后验概率支持向量机的启发, 但是与他们还是完全不同的. 首先, 研究对象是不同的, 本文主要研究的是高斯过程模型, 而文献 [24-25] 主要研究的是支持向量机; 其次, 研究动机也是不同的, 本文的研究动机主要是解决传统高斯过程分类算法的计算复杂度高的问题, 而后者的目的是处理数据的不平衡问题. 下面对后验概率高斯过程分类算法进行详细介绍.

1 模型建立

本节将以二分类问题为例来描述本文算法. 令 $\mathcal{X} = R^d$ 表示样本的特征空间; $\mathcal{Y} = \{w_1, w_2\}$ 表示样本的标签集合, 为了计算方便, 本文取 $w_1 = +1$, $w_2 = -1$; $D = \{(x_i, y_i)|i = 1, 2, \cdots, n\}$ 表示训练集, $x_i \in \mathcal{X}$ 表示第 i 个样本的特征向量, $y_i \in \mathcal{Y}$ 为样本 x_i 的标签, 如果 $y_i = +1$, 则表示 x_i 是正样本, 否则 x_i 是负样本; $x_* \in \mathcal{X}$ 表示待预测样本. 分类学习的任务是利用训练集 D 在特征空间 $\mathcal{X}$ 上建立一个能输出待预测样本 x_* 的正确标签的函数 $f(x)$. 为了表述方便, 下面将采用 x 表示任何一个样本, y 表示 x 的标签.

传统高斯过程分类模型的基本思想是: 首先, 假设潜变量函数 $f(x)$ 服从高斯过程分布 (即 $f(x) \sim \mathcal{GP}(m(x), k(x, x'))$); 然后, 定义一个S形似然函数 $p(y|f, x) = \mathrm{sig}(y \cdot f(x))$(若采用 Logistic 函数作为S形函数, 则有

$$p(y|f, x) = \frac{1}{1 + \mathrm{e}^{-y \cdot f(x)}};$$

最后, 利用贝叶斯公式得到 $f(x)$ 的后验概率

$$p(f|y, x) = \frac{p(y|f, x) \cdot p(f|x)}{\int p(y|f, x) \cdot p(f|x) \mathrm{d}f}.$$

由于似然函数 $p(y|f, x)$ 是一个S形函数, 不能得到 $p(f|y, x)$ 的分析表达式. 传统高斯过程分类算法定义S形似然函数的主要原因是: 分类问题的样本标签 y 的取值是离散的, 而潜变量函数 $f(x)$ 的取值是连续的; 为了建立 y 与 $f(x)$ 的联系, 人们想到通过建立 $f(x)$ 与 y 发生的概率之间的联系来间接地建立 y 与 $f(x)$ 间的联系, 这就需要一个S形函数来将 $f(x)$ 的值变换到区间 [0,1] 上. 假设不仅已知训练数据集中的每个样本 x_i 的标签 y_i, 还已知 x_i 的标签是 y_i 的概率 (即 $p(y_i|x_i)$), 那么期望得到的预测函数就是满足条件

$$\{p(y_i|x_i) \equiv p(y_i|f, x_i)|i = 1, 2, \cdots, n\}$$

的潜变量函数 $f(x)$; 而寻找满足条件

$$\{p(y_i|x_i) \equiv p(y_i|f, x_i)|i = 1, 2, \cdots, n\}$$

的潜变量函数 $f(x)$ 就等价于寻找满足条件

$$\left\{f(x_i) \equiv y_i \cdot \ln\left(\frac{1}{p(y_i|x_i)} - 1\right)|i = 1, 2, \cdots, n\right\} \quad (1)$$

的潜变量函数 $f(x)$, 这里取 $\mathrm{sig}(t) = \dfrac{1}{1 + \mathrm{e}^{-t}}$, 可以看出式 (1) 显然是一个回归问题; 因此, 只要知道概率 $p(y_i|x_i)$, 便可通过求解回归问题

$$\Big\{f(x_i)\equiv y_i\cdot\ln\Big(\frac{1}{p(y_i|x_i)}-1\Big)|i=1,2,\cdots,n\Big\}$$

得到原始分类问题的一个预测函数 $f(x)$, 从而可避免定义 S 形似然函数所带来的高计算复杂度问题. 下面详细介绍后验概率高斯过程分类算法的各个模块.

1.1 估计后验概率 $p(y_i|x_i)$

如果已知类条件先验概率 $p(x_i|w_j)(j=1,2)$ 和类先验概率 $p(w_j)(j=1,2)$, 则由贝叶斯公式

$$p(w_j|x_i)=\frac{p(x_i|w_j)p(w_j)}{p(x_i|w_1)p(w_1)+p(x_i|w_2)p(w_2)},\ j=1,2,$$

可以很容易地计算出样本 x_i 的后验概率

$$p(y_i|x_i)=\begin{cases}p(w_1|x_i),\ y_i=w_1;\\ p(w_2|x_i),\ y_i=w_2.\end{cases}\tag{2}$$

本文采用 Parzen 窗方法来估计 $p(x_i|w_j),\ j=1,2$. Parzen 窗方法是一种非参数密度估计方法[26], 其基本思想是利用一定范围内各样本点密度的平均值对总体概率密度进行估计. 由于该方法具有坚实的理论基础和许多优良的性能, 现已广泛应用于各种应用问题, 当然它也具有对训练样本的需求量较大和存在维数灾难问题等缺点. 因为本文的重点不是如何估计 $p(x_i|w_j)$, 所以将直接采用最基本的 Parzen 窗方法来估计 $p(x_i|w_j)$, 有兴趣的读者可作进一步改进.

令 $\mathcal{X}_k^{w_j}(x_i)$ 表示 x_i 在训练集中的类别为 w_j 的 l 个近邻点组成的集合, 将 $p(x_i|w_j)$ 定义为

$$p(x_i|w_j)=\frac{1}{l}\sum_{x\in\mathcal{X}_l^{w_j}(x_i)}\psi_i(x),\tag{3}$$

其中

$$\psi_i(x)=(2\pi\theta^2)^{-d/2}\exp\Big(-\frac{(x-x_i)^{\mathrm{T}}(x-x_i)}{2\theta^2}\Big).$$

这里: $\exp(\cdot)$ 为以 e 为底的指数函数, θ 为一固定参数.

令 n_j 表示训练集中标签为 $w_j(j=1,2)$ 的样本个数, 则 $p(w_j)(j=1,2)$ 可由 n_j 估计得到, 即

$$p(w_j)=n_j/n.\tag{4}$$

在计算 $p(w_j|x_i)$ 时可能会存在"x_i 的标签为 w_1, 而 $p(w_1|x_i)<0.5$ (或者 x_i 的标签为 w_2, 而 $p(w_2|x_i)<0.5$)"的情况. 当遇到这种情况时, 本文强制地令 $p(w_j|x_i)=0.5+\varepsilon_1$, 其中 ε_1 是一个正常数, 在本文的实验部分, 取 $\varepsilon_1=0.01$.

1.2 学习潜变量函数 $f(x)$

根据上一节所得样本 x_i 的后验概率 $p(y_i|x_i),\ i=1,2,\cdots,n$, 可将原始训练数据集变换为如下数据集:

$$S=\Big\{(x_i,z_i)|z_i=-y_i\cdot\ln\Big(\frac{1}{p(y_i|x_i)}-1\Big),\ i=1,2,\cdots,n\Big\}.\tag{5}$$

因为当 $p(y_i|x_i)=1$ 时, $\ln\Big(\frac{1}{p(y_i|x_i)}-1\Big)$ 是无穷大, 所以在实际计算时, 对于 $p(y_i|x_i)=1$ 的情形, 本文将强制地取 $p(y_i|x_i)=1-\varepsilon_2$, 在本文的实验部分, 取 $\varepsilon_2=0.01$. 下面讨论如何利用变换后的训练数据集学习潜变量函数 $f(x)$.

假设 $f(x)$ 服从如下零均值高斯过程分布:

$$f(x)\sim\mathcal{GP}(0,k(x,x')),\tag{6}$$

其中 $k(x,x')$ 表示协方差函数. 在仿真实验中将使用如下形式的协方差函数:

$$k(x,x')=\alpha_1\cdot\exp\Big(-\frac{||x-x'||^2}{\alpha_2}\Big),\tag{7}$$

其中 α_1 和 α_2 是两个正参数.

令 f_i 表示 $f(x)$ 在样本 x_i 的值, 即 $f_i=f(x_i)$; $F=[f_1,f_2,\cdots,f_n]^{\mathrm{T}}$ 表示由 $f(x)$ 在整个样本集 $X=\{x_1,x_2,\cdots,x_n\}$ 上的值构成的向量. 根据式 (6) 可得到 F 的先验概率

$$p(F|X)=\mathcal{N}(F|0,K).\tag{8}$$

其中: K 为协方差函数 $k(x,x')$ 在 X 上的值构成的矩阵, 即 K 的第 i 行 j 列元素为 $k(x_i,x_j)$. 同时, F 与 f_* 间的联合先验概率可表示为

$$p(F,f_*|X,x_*)=\mathcal{N}\Big(\begin{bmatrix}f_*\\F\end{bmatrix}\Big|0,\begin{bmatrix}K_{**}&K_*^{\mathrm{T}}\\K_*&K\end{bmatrix}\Big).\tag{9}$$

其中: $f_*=f(x_*)$ 表示 $f(x)$ 在待预测样本 x_* 上的值; $K_{**}=k(x_*,x_*)$; K_* 为一个列向量, 它的第 i 个元素为 $k(x_*,x_i)$. 根据式 (9) 可得到 f_* 的条件先验概率

$$p(f_*|F,X,x_*)=\mathcal{N}(f_*|K_*^{\mathrm{T}}K^{-1}F,K_{**}-K_*^{\mathrm{T}}K^{-1}K_*).\tag{10}$$

假设样本的观察值 z_i 带有零均值高斯噪声, 则似然函数 $p(z_i|f_i)$ 可定义为

$$p(z_i|f_i)=\mathcal{N}(z_i|f_i,\sigma^2),\tag{11}$$

其中 σ^2 表示高斯噪声的方差. 假设样本之间是相互独立的, 则联合似然函数 $p(Z|F)$ 可表示为

$$p(Z|F)=\prod_{i=1}^{n}p(z_i|f_i)=\mathcal{N}(Z|F,\sigma^2I).\tag{12}$$

其中: $Z=[z_1,z_2,\cdots,z_n]^{\mathrm{T}}$, I 为单位矩阵.

由先验概率 (8) 和似然函数 (12), 可利用贝叶斯公式得到 F 的后验概率

$$p(F|X,Z)=\frac{p(Z|F)p(F|X)}{p(Z|X)}=\mathcal{N}\Big(F|(I+\sigma^2K^{-1})^{-1}Z,\Big(\frac{1}{\sigma^2}I+K^{-1}\Big)^{-1}\Big),\tag{13}$$

其中 $p(Z|X)=\int p(Z|F)p(F|X)\mathrm{d}F$ 为边缘似然函数. 本文可通过最大化边缘似然 $p(Z|X)$ 来学习参数 σ, 即

$$\hat{\sigma}=\arg\max_{\sigma}p(Z|X)=\arg\max_{\sigma}\log p(Z|X)=\arg\max_{\sigma}\Big\{-\frac{1}{2}(Z^{\mathrm{T}}(K+\sigma^2I)^{-1}Z+\ln|2\pi(K+\sigma^2I)|)\Big\}.\tag{14}$$

利用 f_* 的条件先验概率(10)和 F 的后验概率(13)得到 f_* 的后验概率公式

$$p(f_*|X,Z,x_*)=\int p(f_*|F,X,x_*)p(F|X,Z)\mathrm{d}F=\mathcal{N}(f_*|K_*^{\mathrm{T}}(K+\sigma^2I)^{-1}Z,K_{**}-K_*^{\mathrm{T}}(K+\sigma^2I)^{-1}K_*). \tag{15}$$

式(15)即为所要学习的潜变量函数 $f(x)$ 的表达式.

1.3 预 测

利用 f_* 的后验概率公式(15), 估计待预测样本 x_* 的标签是 $+1$ 的概率为

$$p(y_*=+1|D,x_*)=\int\frac{1}{1+\exp(-f_*)}p(f_*|X,Z,x_*)\mathrm{d}f_*. \tag{16}$$

由于上式的积分中包含了 Logistic 函数 $\dfrac{1}{1+\exp(\cdot)}$, 并不能得到其分析表达式, 本文将利用文献[27]中的方法计算它的一个近似表达式, 即

$$p(y_*=+1|D,x_*)\approx\frac{1}{1+\exp\Big(-\Big(a/\sqrt{1+\pi b^2/8}\Big)\Big)}. \tag{17}$$

其中: a,b 分别表示 $f_*|X,Z,x_*$ 的均值和方差.

到目前为止, 本文已经详细介绍了算法的各个主要模块. 下面给出算法的具体流程:

训练

输入: 训练样本集 $D=\{(x_i,y_i)|i=1,2,\cdots,n\}$, 参数 $l,\theta,\alpha_1,\alpha_2$.

Step 1: 利用式(2)计算 $p(y_i|x_i),i=1,2,\cdots,n$;

Step 2: 利用式(5)变换训练数据集, 并得到 Z;

Step 3: 计算协方差矩阵 K, 并对其作特征分解 $K=P^{\mathrm{T}}\Lambda P$;

Step 4: 利用式(14)学习参数 σ;

Step 5: 计算 $(K+\sigma^2I)^{-1}=P^{\mathrm{T}}(\Lambda+\sigma^2I)^{-1}P$.

输出: $Z,(K+\sigma^2I)^{-1}$.

预测

输入: 待测样本 x_*, Z, $(K+\sigma^2I)^{-1}$.

Step 1: 计算 K_{**}, K_*;

Step 2: 计算

$$K_*^{\mathrm{T}}(K+\sigma^2I)^{-1}Z,\ K_{**}-K_*^{\mathrm{T}}(K+\sigma^2I)^{-1}K_*;$$

Step 3: 利用式(17)计算 $p(y_*=+1|D,x_*)$.

输出: $p(y_*=+1|D,x_*)$.

从以上算法流程可以看出: 在训练阶段, 本文算法的主要计算量来自对协方差矩阵 K 进行特征分解, 即计算复杂度至多为 $O(n^3)$; 在测试阶段, 算法的计算复杂度为 $O(n^2)$. 因此, 本文算法的计算复杂度明显要比 LA、EP 和 VB 等(训练阶段的计算复杂度为 $O(\ln^3)$)传统高斯过程分类算法低.

2 仿真实验

本节将在取自于 UCI 数据库[28]的 16 个基准数据集上验证本文算法的性能. 这些数据集分别来自于金融、医疗和物理等 6 个不同的科学领域, 各个数据集的名称、样本个数等其他详细信息如表 1 所示. 为了验证算法的性能, 同时将本文算法与 LA[9], EP[10], VB[11]和 TGPs[23] 四种已有的高斯过程分类算法进行比较, 在实验过程中, 本文以高斯核函数(7)作为各个算法的协方差函数, 利用五折交叉验证法选择算法的参数.

表 1 基准数据集的详细信息

数据集名称	正样本个数	负样本个数	特征维数	所属领域
Arrhythmia	427	25	278	医疗
Australian	383	307	14	金融
BUPA	145	200	6	医疗
Crime	1844	150	100	社会
Sonar	111	97	60	物理
Ecoli	301	35	7	生物
Heart-C	164	139	13	医疗
Heart-S	150	120	13	医疗
Hepatitis	32	123	19	医疗
Ionosphere	126	225	34	物理
Libras	336	24	90	物理
Spectrometer	486	45	93	物理
Oil	896	41	49	环境
Pima	500	268	8	医疗
WDBC	212	357	30	医疗
WPBC	47	151	33	医疗

表 2 列出了各个算法在每个数据集上的预测精度, 所有结果都是利用 10 次十折交叉验证法计算得到的平均结果, 其中每个数据集上的最好结果用黑体加粗显示. 由表 2 可以看出, 本文算法在一半的数据集上取得了最好的预测精度, TGPs 算法在 4 个数据集上取得了最好的预测精度, 其他 3 个算法分别在 3 个数据集上取得了最好的预测精度. 在没有取得最好预测精度的数据集上, 本文算法只有在 Heart-C 数据集上的预测精度与 TGPs 算法相当而明显不如 LA、EP 和 VB 三个算法, 在其他 7 个数据集上与 LA、EP 和 VB 算法的结果所差无几. 因此, 从总体上看, 本文算法要优于其他几个算法.

为了进一步明确各个算法的优劣, 本文利用 t 检验方法对这些算法在各个数据集上定义了一个偏序关系"$\succ$", 即如果 t 检验方法的检验结果显示在某个数据集上算法 A 的预测精度在统计上高于算法 B 的预测精度, 则在该数据集上记作 A $\succ$ B, 其中 t 检验方法的显著性水平取 5%. 表 3 为本文算法与 EP、LA、TGPs 和 VB 四种算法在不同基准数据集上的相对性

表 2　各个算法在基准数据集上的预测精度 (mean±std)　　%

数据集名称	算法名称				
	本文算法	EP	LA	TGPs	VB
Arrhythmia	**94.47±0.00**	94.25±0.16	94.25±0.16	**94.47±0.00**	94.29±0.10
Australian	**86.46±0.19**	86.29±0.35	86.35±0.26	86.35±0.19	86.32±0.30
BUPA	72.46±0.29	**72.93±0.78**	72.81±0.80	69.91±1.67	72.75±0.74
Crime	93.87±0.09	93.97±0.07	94.08±0.09	93.50±0.17	**94.12±0.13**
Ecoli	**93.51±0.13**	92.56±0.30	92.68±0.16	92.80±0.49	92.68±0.16
Heart-C	81.58±0.28	83.30±0.50	**83.37±0.38**	81.45±0.54	83.30±0.50
Heart-S	**84.22±0.50**	82.89±0.31	82.89±0.31	82.89±0.41	82.96±0.26
Hepatitis	**85.70±0.54**	84.52±0.79	84.13±1.68	85.03±0.84	84.39±0.84
Ionosphere	92.36±0.24	91.28±0.52	90.48±0.59	**93.45±0.35**	90.83±0.62
Libras	97.94±0.46	97.94±0.46	**98.06±0.39**	97.72±0.53	98.00±0.36
Oil	**95.62±0.00**	**95.62±0.00**	**95.62±0.00**	**95.62±0.00**	**95.62±0.00**
Pima	**78.13±0.32**	77.24±0.17	77.27±0.17	76.41±0.34	77.24±0.17
Sonar	88.56±0.40	89.13±0.73	89.23±0.73	**90.77±0.63**	88.94±1.12
Spectrometer	95.44±0.21	93.56±0.08	95.89±1.22	95.29±0.13	**96.61±0.35**
WDBC	97.34±0.15	**97.53±0.12**	97.50±0.19	97.43±0.01	97.43±0.16
WPBC	**81.21±0.66**	80.91±0.42	80.51±0.68	81.11±0.28	80.61±0.28

能比较结果. 其中: P, E, L, T 和 V 分别代表本文算法, EP 算法, LA 算法, TGPs 算法和 VB 算法的相对性能. 从统计意义上看, 所有算法在 Australian、WDBC、Libras 和 Oil 四个数据集上的预测结果都是一样的; 本文算法在 Heart-S, Hepatitis, Ionosphere, Arrhythmia 和 Ecoli 五个数据集上一致优于 LA、EP 和 VB 算法, 而在 Heart-C 和 Crime 数据集上劣于这 3 个算法. 为了给出一个整体性能的排序, 在表 3 的最后一行同时给出了每个算法的一个总体得分, 得分的计算方法为: 如果在某个数据集上有 A ≻ B, 则算法 A 将得到 1 分, 而算法 B 将被扣掉 1 分. 从各个算法的得分可以看出, 本文算法在整体上优于其他算法.

表 3　各个算法在基准数据集上的相对性能

数据集名称	性能比较结果
Arrhythmia	P≻L, P≻E, P≻V, T≻L, T≻E, T≻V
Australian	N/A
BUPA	P≻T, L≻T, E≻T, V≻T
Crime	P≻T, L≻P, L≻T, L≻E, E≻P, E≻T, V≻P, V≻T, V≻E
Ecoli	P≻T, P≻L, P≻E, P≻V
Heart-C	L≻T, E≻T, V≻T, L≻P, E≻P, V≻P
Heart-S	P≻T, P≻L, P≻E, P≻V
Hepatitis	P≻L, P≻E, P≻V
Ionosphere	P≻L, P≻E, P≻V, T≻P, T≻L, T≻E,T≻V, E≻L
Libras	N/A
Oil	N/A
Pima	P≻T, P≻L, P≻E, P≻V, L≻T, E≻T,V≻T
Sonar	T≻P, T≻L, T≻E, T≻V
Spectrometer	P≻E, T≻E, L≻E, V≻P, V≻T, V≻E
WDBC	N/A
WPBC	P≻V, T≻V
得分	本文算法(15) > LA(−1) = VB(−1) > TGPs(−5) > EP(−8)

3　结　论

考虑到传统高斯过程分类算法在逼近潜变量函数的后验概率时会遭遇计算复杂度高的问题, 本文通过引入训练数据的后验概率, 提出了一种新的高斯过程分类算法. 在该算法中, 可直接分析地得到潜变量函数的后验概率表达式, 从而避免传统方法逼近潜变量函数的后验概率时所遭遇的困难. 在 16 个来自 UCI 数据库的基准数据集上的仿真实验结果表明, 本文算法优于已有算法. 下一步将研究如何将该算法推广到多类分类问题, 以及如何有效地学习算法的参数, 从而使该算法更具竞争力.

参考文献(References)

[1] Shawe-Taylor J, Cristianini N. Kernel methods for pattern analysis[M]. Cambridge: Cambridge University Press, 2004.

[2] Ji Y, Sun S. Multitask multiclass support vector machines: Model and experiments[J]. Pattern Recognition, 2013, 46(3): 914-924.

[3] Shawe-Taylor J, Sun S. A review of optimization methodologies in support vector machines[J]. Neurocomputing, 2011, 74(17): 3609-3618.

[4] Rasmussen C, Williams K. Gaussian process for machine learning[M]. Cambridge: The MIT Press, 2006.

[5] Neal R. Bayesian learning for neural networks[J]. Lecture Notes in Statistics, 1996, 118.

[6] Sun S, Xu X. Variational inference for infinite mixtures of Gaussian processes with applications to traffic flow prediction[J]. IEEE Trans on Intelligent Transportation Systems, 2011, 12(2): 466-475.

[7] Sun S. Infinite mixtures of multivariate Gaussian processes[C]. Proc of the Int Conf on Machine Learning and Cybernetics. Tianjin, 2013: 1-6.

[8] Ranganathan A, Yang M, Ho J. Online sparse Gaussian process regression and its applications[J]. IEEE Trans on Image Processing, 2011, 20(2): 391-404.

[9] Williams C, Barber D. Bayesian classification with Gaussian processes[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 1998, 20(12): 1342-1351.

[10] Kim H, Ghahramani Z. Bayesian Gaussian process classification with the EM-EP algorithm[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2006, 28(12): 1948-1959.

[11] Opper M, Archambeau C. The variational Gaussian approximation revisited[J]. Neural Computation. 2009, 21(3): 786-792.

[12] Xu Z, Kersting K, Tresp V. Multi-relational learning with gaussian processes[C]. The 21st Int Joint Conf on Artificial Intelligence. Pasadena, 2009: 1309-1314.

[13] Engel Y, Mannor S, Meir R. Reinforcement learning with Gaussian processes[C]. Proc of the 22nd Int Conf on Machine Learning. Bonn, 2005: 201-208.

[14] Lawrence N, Jordan M. Semi-supervised learning via Gaussian processes[C]. Advances in Neural Information Processing Systems. Vancouver, 2005: 753-760.

[15] Guiver J, Snelson E. Learning to rank with softrank and gaussian processes[C]. Proc of the 31st Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval. Singapore, 2008: 259-266.

[16] Bonilla E, Chai K, Williams C. Multi-task Gaussian process prediction[C]. Advances in Neural Information Processing Systems. Vancouver, 2008: 253-260.

[17] He J, Gu H, Wang Z. Bayesian multi-instance multi-label learning using Gaussian process prior[J]. Machine Learning, 2012, 88(1/2): 273-295.

[18] He J, Gu H, Wang Z. Multi-instance multi-label learning based on Gaussian process with application to visual mobile robot navigation[J]. Information Sciences, 2012, 190: 162-177.

[19] 贺建军. 基于高斯过程模型的机器学习算法研究及应用[D]. 大连: 大连理工大学控制科学与工程学院, 2012. (He J J. Research and application of machine learning algorithms based on Gaussian process model[D]. Dalian: School of Control Science and Engineering, Dalian University of Technology, 2012.)

[20] Gibbs M, MacKay D. Variational Gaussian process classifiers[J]. IEEE Trans on Neural Networks, 2000, 11(6): 1458-1464.

[21] Opper M, Winther O. Mean field methods for classification with Gaussian processes[C]. Advances in Neural Information Processing Systems. Denver, 1999: 309-315.

[22] Barber D, Williams CKI. Gaussian processes for Bayesian classification via hybrid Monte Carlo[C]. Advances in Neural Information Processing Systems. Denver, 1997: 340-346.

[23] He J, Gu H, Jiang S. Twin Gaussian processes for binary classification[C]. IEEE Int Conf on Data Mining. Vancouver, 2011: 1074-1079.

[24] Tao Q, Wu G, Wang F, et al. Posterior probability support vector machines for unbalanced data[J]. IEEE Trans on Neural Networks, 2005, 16(6): 1561-1573.

[25] Gonen M, Tanugur A, Alpaydm E. Multiclass posterior probability support vector machines[J]. IEEE Trans on Neural Networks, 2008, 19(1): 130-139.

[26] Duda R, Hart P, Stork D. Pattern classification[M]. 2nd ed. New York: Wiley, 2001: 164-174.

[27] Maragakis P, Ritort F, Bustamante C, et al. Bayesian estimates of free energies from nonequilibrium work data in the presence of instrument noise [J]. The J of Chemical Physics, 2008, 129(2): 024102.

[28] Frank A, Asuncion A. UCI machine learning repository[EB/OL].[2010]. http://archive.ics.uci.edu/ml.

(责任编辑: 滕 蓉)