文章编号: 1000-0887(2009)01-0090-11

基于 CCH 的 SVM 几何算法及其应用*

彭新俊[1,2], 王翼飞[3]

(1. 上海师范大学 计算数学系, 上海 200234;
2. 科学计算上海高校重点实验室, 上海 200234;
3. 上海大学 数学系, 上海 200444)

摘要: 支持向量机(support vector machine(SVM))是一种数据挖掘中新型机器学习方法. 提出了基于压缩凸包(compressed convex hull(CCH))的 SVM 分类问题的几何算法. 对比简约凸包(reduced convex hull(RCH)), CCH 保持了数据的几何体形状, 并且易于得到确定其极点的充要条件. 作为 CCH 的实际应用, 讨论了该几何算法的稀疏化方法及概率加速算法. 数值试验结果表明所讨论的算法可降低核计算并取得较好的性能.

关 键 词: 支持向量机; 压缩凸包; 核参数; 几何方法; 概率加速
中图分类号: O235; TP18 文献标识码: A

引 言

由 Vapnik 等提出的支持向量机(support vector machine(SVM))[1-3]是一种全新的解决模式识别问题的方法. 与其它分类方法相比较, 如人工神经网络(artificial neural networks(ANN))[4], SVM 具有很多优点, 主要有: 1) SVM 是一个二次规划(quadric programming, (QP))问题, 这保证了解为全局最优解(如果存在解的话); 2) SVM 解的稀疏性确保其具有很好的泛化性; 3) SVM 的实现是基于最小化泛化误差界的结构风险最小化而非经验风险最小化原则; 4) SVM 分类问题具有十分清晰的几何特征. 由于以上优点, 目前 SVM 已经被成功地应用到许多领域[5-8], 并延拓到回归问题中[1-2,9-10].

最近, 一些研究工作者探讨了 SVM 的几何特点. Platt 等[12]讨论了易于实现并具有较快速度的代表性算法——序列最小化(sequential minimal optimization(SMO)). Bennett 等[13]通过对偶表示讨论了 SVM 的几何解释. 结果表明 SVM 等价于求具有最大间隔的分类决策超平面. Keerthi 等[14]最近提出了求解可分情形 SVM 的几何算法, 同时对不可分情形 SVM 引入了变换方法将其变为可分. 然而, 这一变换技术无法推广到 l_1- 范数不可分 SVM 中. 为此, Mavroforakis 等[15]讨论了基于简约凸包(reduced convex hull(RCH))的几何算法. 但将 RCH 引入几何

* 收稿日期: 2008-08-26; 修订日期: 2008-11-17
基金项目: 国家自然科学基金资助项目(30571059); 国家高科技研究发展计划(863)专项资助项目(2006AA02Z190); 上海市重点学科资助项目(S30405)
作者简介: 彭新俊(1980—), 男, 湖南人, 博士(联系人: E-mail: xjpeng@shnu.edu.cn);
王翼飞(1948—), 男, 教授, 博士生导师(E-mail: yifei_wang@shu.edu.cn).

算法也有一些不足,首先,RCH 改变了几何体的形状;其次,仅给出了确定其极点的必要而非充分条件. 本文为解决这些不足引入了压缩凸包(compressed convex hull(CCH))的概念. 相对于RCH,CCH 具有如下优点:1) 不改变原几何体的形状,这保证了 SVM 的解具有更好的泛化性能;2) CCH 极点的数目等于原几何体的极点数. 本文的主要贡献如下:(ⅰ) 分析了 CCH 的理论特性;(ⅱ) 给出了基于竞争凝聚(competitive agglomeration(CA))的鲁棒聚类算法[16]的确定核参数的简单方法;(ⅲ) 为得到稀疏性模型,引入了确定凸包在输入空间中的近似重心方法;(ⅳ) 给出了基于CCH 的SVM 几何算法,并进一步地,为了加快模拟速度,提出了概率加速算法加快计算速度;(ⅴ) 讨论了本文所提出的算法在 UCI 数据库与蛋白质相互作用数据上的应用.

1 预备知识

1.1 SVM 分类

考虑数据集 $D=\{(\boldsymbol{x}_1,y_1),\cdots,(\boldsymbol{x}_l,y_l)\}$ 的分类问题,其中 $\boldsymbol{x}_i\in\mathcal{X}\subset R^m$, $y_i\in\{-1,1\}$. 记 $I^{\pm}$ 为样本的 $y_i=\pm1$ 的对应索引集,并令 $I=I^+\cup I^-$. 简单地说,SVM 是求特征空间中两类训练样本之间的最优分类(最大间隔)平面 $H(\boldsymbol{w},b)$:

$$H(\boldsymbol{w},b):f(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\varphi(\boldsymbol{x})+b=0,$$

其中,$\varphi(\cdot):\mathcal{X}\to\mathcal{H}$ 将 $\mathcal{X}$ 映射到特征空间 $\mathcal{H}$ 中且 $\boldsymbol{w}\in\mathcal{H}$. 根据Mercer定理,可用核 $k(\boldsymbol{u},\boldsymbol{v})$ 表示 $\mathcal{H}$中的内积,即 $k(\boldsymbol{u},\boldsymbol{v})=\varphi(\boldsymbol{u})^{\mathrm{T}}\varphi(\boldsymbol{v})$,如 RBF 核

$$k(\boldsymbol{u},\boldsymbol{v})=\exp\{-\|\boldsymbol{u}-\boldsymbol{v}\|^2/(2\sigma^2)\}.$$

为确定实际应用中的分类超平面,通常需要为每一个样本 $\boldsymbol{x}_i$ 引入松弛变量 ξ_i 以确保约束条件成立,从而 SVM 的模型为:

$$\begin{aligned}&\min\ \frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}+C\sum_{i\in I}\xi_i\\&\text{s.t. } y_i(\boldsymbol{w}^{\mathrm{T}}\varphi(\boldsymbol{x}_i)+b)\geqslant1-\xi_i,\\&\qquad\xi_i\geqslant0,\ i\in I,\end{aligned}$$

其中 $C>0$ 为给定的正则化参数. 该问题是 $\mathcal{H}$中具有线性不等式约束的QP 问题. 引入 Lagrange 系数 α_i,我们得到其对偶:

$$\begin{cases}\max\ \sum_{i\in I}\alpha_i-\frac{1}{2}\sum_{i,j\in I}\alpha_i\alpha_j y_i y_j k(\boldsymbol{x}_i,\boldsymbol{x}_j)\\\text{s.t. } 0\leqslant\alpha_i\leqslant C,\ i\in I,\\\sum_{i\in I}\alpha_i y_i=0\left(\text{或}\sum_{i\in I^-}\alpha_i=\sum_{i\in I^+}\alpha_i\right).\end{cases}\tag{1}$$

求解该问题即可得到特征空间中的分类超平面:

$$f(\boldsymbol{x})=\sum_{i\in I}\alpha_i y_i k(\boldsymbol{x}_i,\boldsymbol{x})+b,$$

其中,$H(\boldsymbol{w},b)$ 中的 $\boldsymbol{w}$ 由Karush-Kuhn-Tucker(KKT)优化条件得到

$$\boldsymbol{w}=\sum_i y_i\alpha_i\varphi(\boldsymbol{x}_i)=\sum_{i\in I^+}\alpha_i\varphi(\boldsymbol{x}_i)-\sum_{i\in I^-}\alpha_i\varphi(\boldsymbol{x}_i).$$

注意到超平面 $H(\boldsymbol{w},b)=H(s\boldsymbol{w},sb)$, $s>0$,这是由于对 $\boldsymbol{w}$ 与 b 的缩放不改变超平面的几何特性. 因此,我们可假设问题(1)中约束 $\sum_{i\in I^+}\alpha_i=\sum_{i\in I^-}\alpha_i=1$ 成立. 在该假设下我们得到SVM 的直观几何解释:可分情形 SVM 等价于求由两类训练样本构成的凸包间的最近点对,最

大间隔超平面为这两个点之间的中垂平面. 而不可分情形 SVM 等价于求两个 RCH 之间的最近点对. 关于 SVM 详细的几何解释可见文献[15, 17-19].

1.2 简约凸包(RCH)

定义 1 设集合 X 由 k 个不同点组成. 对实数 $0<\mu<1$, X 的简约凸包 RCH(X, μ) 定义为

$$\mathrm{RCH}(X,\mu)=\left\{\boldsymbol{x}:\boldsymbol{x}=\sum_{i=1}^{k}a_i\boldsymbol{x}_i,\ \boldsymbol{x}_i\in X,\ \sum_{i=1}^{k}a_i=1,\ 0\leqslant a_i\leqslant\mu\right\}.$$

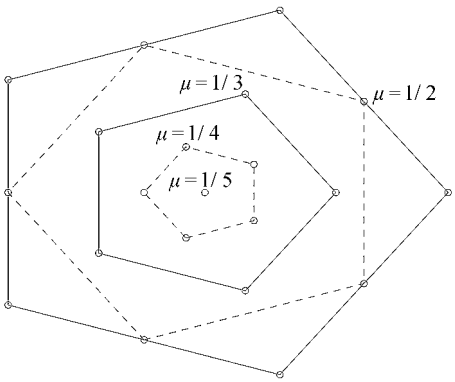

图 1 不同 μ 值的 RCH 示意图

图 1 给出了具有不同 μ 值的 RCH 示意图. 显见, 对于不可分情形 SVM, 可通过合适的 μ 值将两个部分重叠的凸包转化为线性可分情形.

命题 1[15] 点集 X 的简约凸包 RCH(X, μ) 的极点系数 $a_i\in S=\{0, \mu, 1-\lfloor 1/\mu\rfloor\mu\}$, 即 RCH$(X, \mu)$ 的极点由 X 中 $m=\lceil 1/\mu\rceil$ 个不同点简约凸组合而成. 进一步, 若 $1/\mu=\lceil 1/\mu\rceil$, 则 $a_i=\mu$, $i=1, \ldots, m$, 否则 $a_i=\mu$, $i=1, \ldots, m-1$, $a_m=1-\lfloor 1/\mu\rfloor\mu$.

命题 1 给出了 RCH 的极点的必要而非充分条件, 满足该条件的点数目远大于极点数目. 因而在求最近点对的过程中不可能一一检查所有可能极点. 事实上, 在几何算法中, 我们并不需要逐一考虑每一个极点, 而仅需要计算所有正负训练样本的投影, 并选择前 $\lceil 1/\mu\rceil$ 个具有较小投影的正与负训练样本采用以上方式进行凸组合. 具体过程详见文献[15].

2 压缩凸包(CCH)

尽管 RCH 能将不可分问题转化为可分情形, 但存在一些不足. 首先, 它改变训练集凸包的几何形状. 其次, 仅给出了确定其极点的必要而非充分条件. 为此, 我们引入一个新的压缩凸包 CCH 的概念:

定义 2 设集合 X 由 k 个不同点组成. 对实数 $0<\lambda<1$, X 的压缩凸包 CCH(X, λ) 定义为

$$\mathrm{CCH}(X,\lambda)=\left\{\boldsymbol{x}:\boldsymbol{x}=\sum_{i=1}^{k}a_i\hat{\boldsymbol{x}}_i,\ \hat{\boldsymbol{x}}_i=(1-\lambda)\boldsymbol{x}_c+\lambda\boldsymbol{x}_i,\ 0\leqslant a_i\leqslant 1,\ \sum_{i=1}^{k}a_i=1,\ \boldsymbol{x}_c=\frac{1}{k}\sum_{i=1}^{k}\boldsymbol{x}_i\right\}. \tag{2}$$

由 CCH 的定义, CCH(X, λ) 中的每一点都是 X 中的对应点朝其重心 $\boldsymbol{x}_c$ 方向压缩. 显见, 当 $\lambda=1$ 时, CCH$(X, 1)$ 即为 X 的凸包 conv(X), 即无任何压缩. 当 $\lambda=0$ 时, CCH$(X, 0)$ 退化为 X 的重心 $\boldsymbol{x}_c$ 构成的单点集. 因而, 给定合适的参数 λ 可将部分重叠的凸包转化为可分. 图 2 给出了不同 λ 值下 X 的 CCH 示意图. 显见, 由于 CCH 可基本保持 X 的凸包的形状, 从而能保持训练数据的许多特征.

下面将讨论 CCH 的一些性质, 这些性质有助于我们理解 CCH 并构建新的几何算法.

命题 2 点集 X 的压缩凸包 CCH(X, λ) 可等价地表示为

$$\mathrm{CCH}(X,\ \lambda) = \left\{ \boldsymbol{x}:\ \boldsymbol{x} = \sum_{i=1}^{k} a_i \boldsymbol{x}_i,\ a_i \geqslant \frac{1-\lambda}{k},\ \sum_{i=1}^{k} a_i = 1 \right\}.$$

并且, X 的压缩集$\{\hat{\boldsymbol{x}}_1, \cdots, \hat{\boldsymbol{x}}_k\}$的重心仍为$X$ 的重心$\boldsymbol{x}_c$.

证明 首先计算 X 的压缩集的重心

$$\hat{\boldsymbol{x}}_c = \frac{1}{k}\sum_{i=1}^{k}\hat{\boldsymbol{x}}_i = \frac{1}{k}\sum_{i=1}^{k}[(1-\lambda)\boldsymbol{x}_c + \lambda\boldsymbol{x}_i] = \boldsymbol{x}_c.$$

即得结论.

其次, 记 $C(X,\ \lambda)$ 为

$$C(X,\ \lambda) = \left\{ \boldsymbol{x}:\ \boldsymbol{x} = \sum_{i=1}^{k} a_i\boldsymbol{x}_i,\ a_i \geqslant \frac{1-\lambda}{k},\ \sum_{i=1}^{k} a_i = 1 \right\}.$$

对任给的 $\boldsymbol{x} \in \mathrm{CCH}(X,\ \lambda)$, 有

$$\boldsymbol{x} = \sum_{i=1}^{k} a_i \hat{\boldsymbol{x}}_i = \sum_{i=1}^{k}\left(\frac{1-\lambda}{k} + \lambda a_i\right)\boldsymbol{x}_i.$$

令 $\beta_i = (1-\lambda)/k + \lambda a_i$, 则有

$$\beta_i \geqslant (1-\lambda)/k,\ \sum_{i=1}^{k}\beta_i = 1,$$

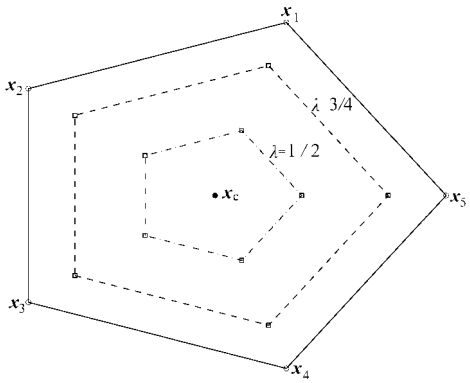

图 2 不同 λ 值的 CCH 示意图

即 $\boldsymbol{x} \in C(X,\ \lambda)$. 反之, 对任给的 $\boldsymbol{x} = \sum_{i=1}^{k} a_i\boldsymbol{x}_i \in C(X,\ \lambda)$, 记 $\zeta_i = (a_i - (1-\lambda)/k)/\lambda$, 则有

$$\zeta_i \geqslant 0,\ \sum_{i=1}^{k}\zeta_i = 1$$

成立, 且

$$\boldsymbol{x} = \sum_{i=1}^{k}\left(\lambda\zeta_i + \frac{1-\lambda}{k}\right)\boldsymbol{x}_i = \sum_{i=1}^{k}\zeta_i\hat{\boldsymbol{x}}_i \in \mathrm{CCH}(X,\ \lambda).$$

命题得证. □

命题 3 对任给 $\boldsymbol{x}_i \in X$, $\boldsymbol{x}_i$ 为其凸包的极点当且仅当其压缩$\hat{\boldsymbol{x}}_i = (1-\lambda)\boldsymbol{x}_c + \lambda\boldsymbol{x}_i$ 为 $\mathrm{CCH}(X,\ \lambda)$ 的极点.

证明 设 $\hat{\boldsymbol{x}}_i = (1-\lambda)\boldsymbol{x}_c + \lambda\boldsymbol{x}_i$ 不是$\mathrm{CCH}(X,\ \lambda)$ 的极点, 但 $\boldsymbol{x}_i$ 为X 的凸包的极点, 即存在 $0 < \alpha < 1$ 及 $\boldsymbol{u},\ \boldsymbol{v} \in \mathrm{CCH}(X,\ \lambda)$ $(\boldsymbol{u} \neq \boldsymbol{v})$ 成立$\hat{\boldsymbol{x}}_i = \alpha\boldsymbol{u} + (1-\alpha)\boldsymbol{v}$, 其中 $\boldsymbol{u} = \sum_j a_j\hat{\boldsymbol{x}}_j$, $\boldsymbol{v} = \sum_j b_j\hat{\boldsymbol{x}}_j$, $a_j,\ b_j \geqslant 0$, $\sum_j a_j = \sum_j b_j = 1$. 则

$$\hat{\boldsymbol{x}}_i = \alpha\boldsymbol{u} + (1-\alpha)\boldsymbol{v} = \sum_j(\alpha a_j + (1-\alpha)b_j)\hat{\boldsymbol{x}}_j =$$

$$(1-\lambda)\boldsymbol{x}_c + \lambda\sum_j(\alpha a_j + (1-\alpha)b_j)\boldsymbol{x}_j.$$

由 $\boldsymbol{x}_i$ 是X 的凸包的极点可得

$$\begin{cases} \alpha a_i + (1-\alpha)b_i = 1, \\ \alpha a_j + (1-\alpha)b_j = 0, \quad j \neq i. \end{cases}$$

若 $a_i \neq b_i$, 则 a_i 与 b_i 中至少有一个小于 1, 从而 $\alpha a_i + (1-\alpha)b_i < 1$, 这与 $\alpha a_i + (1-\alpha)b_i = 1$ 矛盾. 因此有 $a_i = b_i = 1$, $a_j = b_j = 0, j \neq i$. 即 $\boldsymbol{u} = \boldsymbol{v} = \hat{\boldsymbol{x}}_i$, 这与假设矛盾.

反之, 假设 $\boldsymbol{x}_i$ 不是X 的凸包的极点, 但 $\hat{\boldsymbol{x}}_i$ 是$\mathrm{CCH}(X,\ \lambda)$ 的极点, 即存在X 的凸包中的 2 个

不同点 $\boldsymbol{u}=\sum_j a_j\boldsymbol{x}_j$, $\boldsymbol{v}=\sum_j b_j\boldsymbol{x}_j$, 及 $0<\alpha<1$, 使得 $a_j,\ b_j\geqslant 0$, $\sum_j a_j=\sum_j b_j=1$ 与 $\boldsymbol{x}_i=\alpha\boldsymbol{u}+(1-\alpha)\boldsymbol{v}$ 成立. 则

$$\hat{\boldsymbol{x}}_i=(1-\lambda)\boldsymbol{x}_c+\lambda\boldsymbol{x}_i=(1-\lambda)\boldsymbol{x}_c+\lambda\Big[\alpha\sum_j a_j\boldsymbol{x}_j+(1-\alpha)\sum_j b_j\boldsymbol{x}_j\Big]=$$

$$\alpha\Big[(1-\lambda)\boldsymbol{x}_c+\lambda\sum_j a_j\boldsymbol{x}_j\Big]+(1-\alpha)\Big[(1-\lambda)\boldsymbol{x}_c+\lambda\sum_j b_j\boldsymbol{x}_j\Big].$$

因此, 由 $\boldsymbol{u}\neq\boldsymbol{v}$ 可得 $(1-\lambda)\boldsymbol{x}_c+\lambda\sum_j a_j\boldsymbol{x}_j\neq(1-\lambda)\boldsymbol{x}_c+\lambda\sum_j b_j\boldsymbol{x}_j$, 这表明 $\hat{\boldsymbol{x}}_i$ 不是 CCH(X,λ) 的极点, 矛盾. 命题得证. □

基于 CCH 的最优超平面完全决定于由正负训练样本构成的 CCH 的最近点对, 这 2 个点可用各自 CCH 的极点凸表示. 尽管命题 3 指出 CCH 的极点是凸包的对应极点在重心 $\boldsymbol{x}_c$ 方向的凸表示, 但在特征空间中无法简单确定样本集凸包的极点. 文献[15]提供了一种简单的迭代方法, 这里将这一思想直接用于 CCH 中: 计算所有样本点在当前点对连线上的投影, 选择具有最小投影的样本.

3 模型选择与稀疏性控制

3.1 模型选择

SVM 的性能很大程度上依赖于其参数[20-24]. 为了得到每一类的更好的 CCH, 有必要选择合适的核参数. 本节讨论修正 RBF 核 $k(\boldsymbol{u},\boldsymbol{v})=\exp\Big\{-\sum_{t=1}^{m}(u^t-v^t)^2/(2\sigma_t^2)\Big\}$ 中的宽度参数 σ_t 的选择, 其中 $\boldsymbol{u}=(u^1;\ \cdots;\ u^m)\in R^m$.

设 $X=\{\boldsymbol{x}_1,\ \cdots,\ \boldsymbol{x}_k\}$ 为 k 个样本的集合, $\boldsymbol{B}=(\beta_1,\ \cdots,\ \beta_c)$ 表示将样本集划分为 c 个子类, 其中 β_i 表示第 i 个子类. CA 聚类算法[16]优化如下问题:

$$\begin{cases}\min J(B,U,Q)=\sum_{i=1}^{c}\sum_{j=1}^{k}u_{ij}^2\,d^2(\boldsymbol{x}_j,\beta_i)-\eta\sum_{i=1}^{c}\Big(\sum_{j=1}^{k}u_{ij}\Big)^2\\ \text{s.t.}\ \sum_{i=1}^{c}u_{ij}=1,\qquad 0\leqslant j\leqslant k,\end{cases}\tag{3}$$

其中, $d(\boldsymbol{x}_j,\beta_i)$ 是 $\boldsymbol{x}_j$ 与类 β_i 的重心间的距离, $\boldsymbol{U}=[u_{ij}]$ 是样本的模糊 c 划分矩阵, u_{ij} 表示样本 $\boldsymbol{x}_j$ 属于类 β_i 的隶属度, η 为竞争参数. 参数 η 的选择在很大程度上影响目标函数 J, 优化过程中通常采用下述递减方式更新 η[16]:

$$\eta^{(s)}=\eta_0 e^{-s/\tau}\Big\{\Big[\sum_{i=1}^{c}\sum_{j=1}^{k}(u_{ij}^{(s-1)})^2 d^2(\boldsymbol{x}_j,\beta_i)^{(s-1)}\Big]\Big\backslash\Big[\sum_{i=1}^{c}\Big(\sum_{j=1}^{k}u_{ij}^{(s-1)}\Big)^2\Big]\Big\},$$

这里 η_0 为初值, τ 为时间常量, s 为迭代因子. 在聚类过程中, 子类的数目通常是动态变化的.

一旦通过在(3)式中引入 Lagrange 乘子求得 u_{ij}, 每个子类的宽度参数可如下计算:

$$\sigma_t^i=\sqrt{\Big\{\sum_{j=1}^{k}u_{ij}^2(x_j^t-\delta_i^t)^2\Big\}\Big\backslash\Big\{\sum_{j=1}^{k}u_{ij}^2\Big\}},\qquad t=1,\ \cdots,\ m;\ i=1,\ \cdots,\ c,\tag{4}$$

其中, $\delta=(\delta_i^1;\ \cdots;\ \delta_i^m)$ 为类 i 的中心

$$\delta_i=\Big\{\sum_{j=1}^{k}u_{ij}^2\boldsymbol{x}_j\Big\}\Big\backslash\Big\{\sum_{j=1}^{k}u_{ij}^2\Big\},\qquad i=1,\ \cdots,\ c.$$

通常, 子类形状的宽度参数可作为修正 RBF 核的宽度参数[23]. 由于每一子类具有不同的宽度参数, 在实际应用中我们将所有子类的最小宽度值作为核函数的参数.

3.2 稀疏化控制

由CCH的性质可知, 利用CCH得到的分类超平面不再具有SVM的稀疏性. 本节探讨如何稀疏化最优分类超平面的方法, 即如何估计样本的重心点在输入空间中的原象.

对点集 $X = \{x_i: i = 1, \cdots, k\}$, 其特征空间中的重心点可表示为 $(1/k)\sum_{i=1}^{k}\varphi(x_i)$. 我们采用经验风险最小化方法估计该重心点在输入空间中的原象. 即对所有 x_i, 求原始空间中的 x, 使得最小化误差 $L_i = (\|\varphi(x_i) - \varphi(x)\|^2_{\mathcal{H}} - \|\varphi(x_i) - (1/k)\sum_{j=1}^{k}\varphi(x_j)\|^2_{\mathcal{H}})^2$. 对所有 L_i 求和, 得到如下无约束优化问题:

$$\min L = \sum_{j=1}^{k} L_i = \sum_{j=1}^{k}\left(\|\varphi(x_i) - \varphi(x)\|^2_{\mathcal{H}} - \|\varphi(x_i) - \frac{1}{k}\sum_{j=1}^{k}\varphi(x_j)\|^2_{\mathcal{H}}\right)^2. \quad (5)$$

用Newton下降法迭代:

$$x^{\text{new}} = x - (\nabla^2_x L)^{-1}\nabla_x L, \quad (6)$$

其中 $\nabla_x L = \left(\frac{\partial L}{\partial x^1}; \cdots; \frac{\partial L}{\partial x^m}\right)$, $\nabla^2_x L = \left[\frac{\partial^2 L}{\partial x^{t_1}\partial x^{t_2}}\right]_{m\times m}$.

优化问题(5)所得到的最优解可能是局部最优的, 这导致所估计的重心点的原像偏离真实点. 实际上, 可以采用点集的原始空间中的重心点作为优化问题的初值, 这样可得到比较好的重心估计.

4 SVM的几何算法

为简单起见, 记 $z = \varphi(x)$ 为特征空间 $\mathcal{H}$ 中的点, $z_c^{\pm}$ 为正负训练样本在 $\mathcal{H}$ 中的重心, 同时, 记 $\langle z_1, z_2\rangle = k(x_1, x_2)$. 注意到用CCH求解SVM问题时, 正负样本的CCH的极点表示为 $(1-\lambda)z_c^{\pm} + \lambda z_i^{\pm}$, 从而点对 w_1 与 w_2 可分别表示为 $w_1 = (1-\lambda)z_c^{+} + \sum_{i\in I^+} a_i z_i^{+}$ 和 $w_2 = (1-\lambda)z_c^{-} + \sum_{i\in I^-} b_i z_i^{-}$, 其中 $\sum_{i\in I^+} a_i = \sum_{i\in I^-} b_i = \lambda$.

结合核参数选择与稀疏性控制方法, 可将基于CCH的几何SVM算法(CCH-based geometric algorithm for SVM(CCH-GA))表述如下:

1) 模型选择阶段

a) 确定CA聚类算法中的初始聚类数 c、η_0, 时间常量 τ 及阈值 ζ.

b) 估计修正RBF核的宽度参数 σ.

c) 确定两个CCH的参数 λ.

2) 重心估计阶段

d) 估计两个重心点在原始空间中的原象 x_c^{+} 和 x_c^{-};

3) 几何算法阶段

e) 设定初始点对(向量) w_1 与 w_2 为各自压缩凸包的重心, 即

$$w_1 = (1-\lambda)z_c^{+} + \sum_{i\in I^+} a_i \lambda z_i^{+},\ a_i = \lambda / |I^+|,$$

及

$$w_2 = (1-\lambda)z_c^{-} + \sum_{i\in I^-} b_i \lambda z_i^{-},\ b_i = \lambda / |I^-|.$$

f) 停止条件: 求向量

$$\hat{z}_r = \begin{cases} \hat{z}_{i_0}^{+} = (1-\lambda)z_c^{+} + \lambda z_{i_0}^{+} \in \text{CCH}(X^+, \lambda), \\ \hat{z}_{j_0}^{-} = (1-\lambda)z_c^{-} + \lambda z_{j_0}^{-} \in \text{CCH}(X^-, \lambda), \end{cases}$$

使得 $\hat{z}_r = \arg\min_{\hat{z}^+_{i_0}, \hat{z}^-_{j_0}} (m(z^+_{i_0}), m(z^-_{j_0}))$, 其中

$$m(\hat{z}^+_{i_0}) = \frac{\langle z^+_{i_0} - w_2, w_1 - w_2\rangle}{\| w_1 - w_2 \|}, \quad m(\hat{z}^-_{j_0}) = \frac{\langle z^-_{j_0} - w_1, w_2 - w_1\rangle}{\| w_1 - w_2 \|},$$

$$\langle w_1, w_1\rangle = (1-\lambda)^2 + 2(1-\lambda)\sum_{i\in I^+} a_i \langle z^+_c, z^+_i\rangle + \sum_{i,j\in I^+} a_i a_j \langle z^+_i, z^+_j\rangle,$$

$$\langle w_2, w_2\rangle = (1-\lambda)^2 + 2(1-\lambda)\sum_{i\in I^-} b_i \langle z^-_c, z^-_i\rangle + \sum_{i,j\in I^-} b_i b_j \langle z^-_i, z^-_j\rangle,$$

$$\langle w_1, w_2\rangle = (1-\lambda)^2 \langle z^+_c, z^-_c\rangle + \sum_{i\in I^+, j\in I^-} a_i b_j \langle z^+_i, z^-_j\rangle +$$

$$(1-\lambda)\Big[\sum_{i\in I^-} b_i \langle z^+_c, z^-_i\rangle + \sum_{i\in I^+} a_i \langle z^-_c, z^+_i\rangle\Big].$$

若 ε 最优条件 $\| w_1 - w_2 \| - m(\hat{z}_r) < \varepsilon$ 成立, 则向量 $w = w_1 - w_2$ 与 $b = (1/2)(\| w_1 \|^2 - \| w_2 \|^2)$ 为 ε 最优解; 否则, 转 g).

g) 更新: 若 $z_r = \hat{z}^+_{i_0}$, 令 $a^{\text{new}}_{i_0} = q_1\lambda + (1-q_1)a_{i_0}$, $i_0 \in I^+$, $a^{\text{new}}_i = (1-q_1)a_i$, $i \neq i_0$, $i \in I^+$, 其中 $q_1 = \min(1, \langle w_1 - w_2, w_1 - \hat{z}^+_{i_0}\rangle / \| w_1 - \hat{z}^+_{i_0} \|^2)$; 否则 $b^{\text{new}}_{j_0} = q_2\lambda + (1-q_2)b_{j_0}$, $j_0 \in I^-$, $b^{\text{new}}_j = (1-q_2)b_j$, $j \neq j_0$, $j \in I^-$, 其中 $q_2 = \min(1, \langle w_2 - w_1, w_2 - \hat{z}^-_{j_0}\rangle / \| w_2 - \hat{z}^-_{j_0} \|^2)$, 转 f).

由于该算法仅将 RCH-GA 延拓到 CCH 中, 因而具有与 RCH-GA 相同的计算复杂度. 但在模拟中我们可引入某些加速策略. 注意到在算法中, 若置初始点对为两个 CCH 的重心点, 随着迭代的进行, 远离最近点对的样本对 w_1 与 w_2 的影响将逐渐减少. 具体地, 如果一个样本对最近点对没有影响(这里不考虑对重心的影响), 其权重(a_i 或 b_i) 将逐步减少至 0. 因此, 我们引入下面的加速概率 CCH-GA(probabilistic CCH-GA(PCCH-GA)): 在每次迭代中以概率 a_i/λ 或 b_i/λ 随机选择样本, 使得这些样本的权重和占总值的 $100\cdot(1-\alpha)\%$, $0 < \alpha < 1$, 并选择具有最小投影的样本进入更新阶段. 实验中取 $\alpha = 0.05$. 随着迭代的进行, PCCH-GA 可通过降低核计算量来提高算法的运行速度.

注意到 PCCH-GA 对提高 CCH-GA 的计算速度非常有效. 在训练阶段, 大多数样本逐次对最近点对不产生作用, 也就是说这些样本的权重将减少, 并且最终仅有支持向量的权重不为 0, 从而所选取权重达 $100\cdot(1-\alpha)\%$ 的样本的数目越来越少. 换句话说, PCCH-GA 减少核计算量. 另一方面, 该概率加速方案对 RCH-GA 并不如此有效. 这是因为 RCH 的每个极点由 $[1/\mu]$ 个点表示, 而其中权重较小的点仍可能对其极点有影响, 从而导致加速不明显.

5 数值试验

为测试本文所提出的 CCH-GA 及 PCCH-GA 算法的学习性能, 我们在两组数据上进行实验, 第 1 组数据为包括乳腺癌数据集在内的 5 个 UCI Benchmark 数据集, 而第 2 组数据为蛋白质相互作用数据集.

5.1 Benchmark 数据集

我们从 UCI Benchmark 数据库[①]中选用乳腺癌等 5 个数据集(乳腺癌、糖尿病、心脏病、甲

① http://www.ics.uci.edu/~mlearn/MLSummary.html

状腺与铁达尼克)测试并比较 RCH-GA、CCH-GA 与 PCCH-GA 等 3 种算法的性能. 在模拟中,我们采用 10-交叉验证的方法评估这些方法的性能,并采用 CA 聚类算法获取合适的核参数(这里仅考虑用最小估计核参数值实现这些几何算法). 表 1 对比了采用这 3 种算法得到的测试错误和 CPU 运行时间. 结果显示 3 种方法得到与文献[23]类似的结果. 显见,相对于 RCH-GA, CCH-GA 得到更好的分类精度,同时,其运行速度也略快于前者,这主要是由于在 CCH-GA 的每次迭代中仅需考虑一个具有最小投影的样本,从而计算量比 RCH-GA 中考虑 $[1/\mu]$ 个样本的凸组合要小. 比较 3 种算法的训练 CPU 时间可知 PCCH-GA 大幅提高了训练速度,其原因正如前面所述,PCCH-GA 大幅减少了核计算量.

表 1 CCH-GA、PCCH-GA 和 RCH-GA 算法的性能比较

		乳腺癌	糖尿病	心脏病	甲状腺	铁达尼克
RCH-GA	错误率 e/(%)	25.74±4.34	23.47±1.72	15.45±3.23	4.72±2.35	22.45±1.35
	运行时间 t/s	45.34	92.04	40.37	28.78	254.69
CCH-GA	错误率 e/(%)	25.35±4.53	23.16±2.01	15.94±3.07	4.68±2.12	22.59±1.14
	运行时间 t/s	41.32	85.27	36.28	25.65	247.38
PCCH-GA	错误率 e/(%)	26.02±4.74	23.87±2.35	16.42±3.41	4.94±2.23	22.76±1.37
	运行时间 t/s	36.47	78.39	31.97	22.74	224.18

5.2 蛋白质相互作用数据集

本节所用的蛋白质数据来自 SPIN 数据库①,该数据库包含 PDB(protein data bank)数据库中的蛋白质复合物. 为消除极罕见的信号信号子,在复合物中首先去掉所有同源(homodimer)和蛋白酶-抑制子复合物. 其次,为得到具有一般性的分类器,这里仅考虑异源复合物,同样去掉出现多个相互作用的蛋白质链. 另外,删除掉所有的“膜肽”,“小蛋白质”与“卷曲螺旋”. 最后得到下面的 66 个(序列相似性低于 30%)蛋白质: 1aby A, 1agr A E, 1ais B, 1aok A, 1aqd A B, 1aui A B, 1axi A B, 1bpl A B, 1cau A B, 1ebd A C, 1efu A B, 1efv A B, 1fdh G, 1fin A B, 1frv A B, 1gla F G, 1gua A B, 1ibc A B, 1ihf A B, 1jck A B, 1lgb A C, 1mel L, 1mhc A, 1mhl A C, 1mio A B, 1npo A, 1rbl A, 1rlb A E, 1sct A B, 1scu A B, 1tcr A, 1tmc A, 1ttp A B, 1vol A B, 1yrn A B, 2btf A P, 2fgw H, 2pcb A, 2req A B. 在蛋白质相互作用研究工作中,一个残基被定义为表面残基当且仅当其相对 MASA 至少为其正规最大化面积的 25%[25]. 由此我们得到大约 8 300 个表面残基. 一个表面残基被定义为界面残基当且仅当其复合物的可及表面积比单体的可及表面积小至少 1 $\mathring{A}^2$. 根据这一定义得到大约 1/3 的界面残基.

这里采用残基序列谱、进化率与疏水性[26]刻画每个残基的序列信息. 其中残基序列谱可通过单机版 PSI-Blast②计算得到. 进化率的每个输入向量表示残基位置的保守得分. 为刻画目标残基的序列特性,本文引入下面方法: 假设由目标残基 R_i 与其 $2s$ 个序列近邻残基的窗体为

$$R_{i-s}\cdots R_{i-1}R_iR_{i+1}\cdots R_{i+s},$$

其中 R_{i-s+k} 表示该窗体的第 $k+1$ 个残基. 序列谱等相关特性在蛋白质相互作用方面有非常重要的作用. 注意到 $2s$ 个序列最近残基对目标残基的影响是不同的,越接近的残基对 R_i 影

① http://trantor.bioc.columbia.edu/cgi-bin/SPIN/

② http://www.ncbi.nlm.nih.gov/BLAST/

响也越大. 如 $R_{i-1}(R_{i+1})$ 的特性对 R_i 的影响比 $R_{i-s}(R_{i+s})$ 的要大. 因而我们可以根据残基距离目标残基的序列距离刻划序列信息对其影响. 具体地, 通过如下方式反映序列谱、进化率与疏水性对目标残基的影响:

$$\mathrm{Attr}_{i,j} = \frac{1}{2j+1}\sum_{k=-j}^{j}\mathrm{Attr}(R_{i+k}), \qquad j = 0, 1, \cdots, s.$$

经过以上预处理以后, 拼接这些特征得到 $(20+1+1)\times(s+1)$ 维输入向量. 这一方式可反映目标残基周边的序列特性的变化过程, 越接近目标残基的残基特性对目标残基影响也就越多. 实验中考虑 $s = 6$ 时的模拟结果. 注意到仅有大约 30% 的表面残基是界面残基, 为避免负样本过多的情况, 训练中每次随机选取与训练正样本量相同的负样本作为训练负样本. 表 2 给出了这 3 种算法的 5 次 10-交叉验证的平均结果, 这里

$$R_{\mathrm{SE}} = \frac{N_{\mathrm{TP}}}{N_{\mathrm{TP}}+N_{\mathrm{FN}}}\times 100\%,\quad R_{\mathrm{SP}} = \frac{N_{\mathrm{TP}}}{N_{\mathrm{TP}}+N_{\mathrm{FP}}}\times 100\%,\quad R_{\mathrm{AR}} = \frac{N_{\mathrm{TP}}+N_{\mathrm{TN}}}{N}\times 100\%$$

与

$$R_{\mathrm{CC}} = \frac{N_{\mathrm{TP}}\times N_{\mathrm{TN}} - N_{\mathrm{FP}}\times N_{\mathrm{FN}}}{\sqrt{(N_{\mathrm{TP}}+N_{\mathrm{FN}})(N_{\mathrm{TP}}+N_{\mathrm{FP}})(N_{\mathrm{TN}}+N_{\mathrm{FP}})(N_{\mathrm{TN}}+N_{\mathrm{FN}})}}$$

分别表示测试结果的敏感度、特异性、精度与相关系数, 其中 N_{TP}、N_{TN}、N_{FP} 与 N_{FN} 分别表示测试正确的正负样本与测试错误的正负样本数, $N = N_{\mathrm{TP}}+N_{\mathrm{TN}}+N_{\mathrm{FP}}+N_{\mathrm{FN}}$ 表示测试样本的数目. 数值试验结果同样显示 CCH-GA 取得了最好的预测性能, 而 PCCH-GA 的学习时间则最少.

表 2 交叉验证得到的 3 种算法对蛋白质相互作用数据结果比较

	R_{SE}/(%)	R_{SP}/(%)	R_{AR}/(%)	R_{CC}	时间 t/s
RCH-GA	71.24	73.32	72.66	0.4517	548.6
CCH-GA	71.77	76.31	74.68	0.4944	513.4
PCCH-GA	71.51	74.15	72.63	0.4607	457.5

6 结 论

SVM 作为机器学习领域内的一种工具, 由于其具有成熟的数学基础, 它已在许多理论与实际领域中得到相当成功的应用. 本文提出的基于 CCH 的 SVM 几何算法有效地弥补了 RCH-GA 的不足. 为提高算法运行速度, 文章讨论了加速的 PCCH-GA 算法. 同时, 文中讨论了确定核参数与稀疏化方法. 数值试验结果显示本文提出的方法得到比 RCH-GA 算法更好的结果.

致谢 本文作者感谢审稿人与编辑的有益评论与帮助.

[参 考 文 献]

[1] Vapnik V. The Natural of Statistical Learning Theory [M]. New York: Springer, 1995.

[2] Vapnik V. Statistical Learning Theory [M]. New York: Wiley, 1998.

[3] Christianini V, Shawe-Taylor J. An Introduction to Support Vector Machines [M]. Cambridge: Cambridge University Press, 2002.

[4] Ripley B D. Pattern Recognition and Neural Networks [M]. Cambridge: Cambridge University Press, 1996.

[5] El-Naqa I, Yang Y, Wernik M, et al. A support vector machine approach for detection of microclassification [J]. IEEE Trans Med Imag, 2002, **21**(12): 1552-1563.

[6] Joachims T. Text categorization with support vector machines: Learning with many relevant features [A]. In: European Conference on Machine Learning No. 10[C]. **1398**. Chemnitz, Germany: Springer-Verlag, 1998, 137-142.

[7] Osuna E, Freund R, Girosi F. Training support vector machines: an application to face detection[A]. In: Proceedings of the 1997 Conference Computer Vision and Pattern Recognition[C]. Washinton D C: IEEE Computer Society, 1997, 130-136.

[8] Brown M P S, Grundy W N, Lin D, et al. Knowledge-based analysis of microarray gene expression data by using support vector machine[J]. Proc Nat Acad Sci USA, 2000, **97**(1): 262-267.

[9] Mukherjee S, Osuna E, Girosi F. Nonliner prediction of chaotic time series using a support vector machine[A]. In: Proceedings of the 1997 IEEE Workshop[C]. Amelia Island, FL, 1997, 511-520.

[10] Jeng J T, Chuang C C, Su S F. Support vector interval regression networks for interval regression analysis[J]. Fuzzy Sets and Systems, 2003, **138**(2): 283-300.

[11] Zhou D, Xiao B, Zhou H, et al. Global geometric of SVM classifiers[R]. Institute of automation, Chinese Academy of Sciences. Tech Rep AI Lab, 2002.

[12] Platt J. Fast training of support vector machines using sequential minimal optimization[A]. In: Advances in Kernel Method-Support Vector Learning[C]. Cambridge, MA: MIT Press, 1999, 185-208.

[13] Bennett K P, Bredensteiner E J. Geometry in learning[A]. In: Geometry at Work[C]. Washington, DC: Mathematical Association of America, 1998, 132-145.

[14] Keerthi S S, Shevade S K, Bhattacharyya C, et al. A fast iterative nearest point algorithm for support vector machine classifier design[J]. IEEE Trans Neural Netw, 2000, **11**(1): 124-136.

[15] Mavroforakis M E, Theodoridis S. A geometric approach to support vector machine (SVM) classification[J]. IEEE Trans Neural Netw, 2007, **17**(3): 671-682.

[16] Frigui H, Krishnapuram R. A robust competitives clustering algorithm with applications in computer vision[J]. IEEE Trans Pattern Anal Mach Intell, 1999, **21**(5): 450-465.

[17] Bennett K P, Bredensteiner E J. Duality and Geometry in SVM Classifiers[A]. In: Proceedings of the Seventeenth International Conference on Machine Learning[C]. San Mateo, CA: Morgan Kaufmann Publishers Inc, 2000, 57-64.

[18] Crisp D J, Burges C J C. A geometric interpretation of ν-SVM classifiers[A]. In: Advances in Neural Information Processing Systems[C]. Cambridge, MA: MIT Press, 1999, 244-250.

[19] Franc V, Hlavac V. An iterative algorithm learning the maximal margin classifier[J]. Pattern Recognition, 2003, **36**(9): 1985-1996.

[20] Chapelle O, Vapnik V, Bousquet O, et al. Choosing multiple parameters for support vector machines [J]. Mach Learn, 2002, **46**(1): 131-159.

[21] Ayat N E, Cheriet M, Suen C Y. Automatic model selection for the optimization of the SVM kernels [J]. Pattern Recogn Comput Sci, 2005, **38**(10): 1733-1745.

[22] Adankon M M, Cheriet M. Optimizing resources in model selection for support vector machine[J]. Pattern Recognition, 2007, **40**(3): 953-963.

[23] Schittkowshi K. Optimal parameter selection in support vector machine[J]. J Indust Manag Optimi, 2005, **1**(4): 465-476.

[24] Chung K M, Kao W C, Wang L L, et al. Radius margin bounds for support vector machines with the RBF kernel[J]. Neural Comput, 2003, **38**(10): 2643-2681.

[25] Jones S, Thornton J M. Principles of protein-protein interactions[J]. Proc Nat Acad Sci USA, 1996, **93**(1): 13-20.

[26] Glaser F. ConSurf: identification of functional regions in proteins by surface-mapping of phylogenetic

information[J]. Bioinformatics, 2003, **19**(1): 163-164.

CCH-Based Geometric Algorithms for SVM and the Applications

PENG Xin-jun[1,2], WANG Yi-fei[3]

(1. Department of Mathematics, Shanghai Normal University, Shanghai 200234, P.R. China;
2. Scientific Computing Key Laboratory of Shanghai Universities, Shanghai 200234, P.R. China;
3. Department of Mathematics, Shanghai University, Shanghai 200444, P.R. China)

Abstract: The support vector machine (SVM) is a novel machine learning tool in data mining. The geometric approach based on the compressed convex hull (CCH) with a mathematical framework is introduced to solve SVM classification problems. Compared with the reduced convex hull (RCH), CCH preserves the shape of geometric solid for the data set; meanwhile, it is easy to give the necessary and sufficient condition of determining its extreme points. As the practical applications of CCH, spare and probabilistic speed-up geometric algorithms were developed. Results of some numerical experiments show that the proposed algorithms can reduce the kernel evaluation and display nice performances.

Key words: support vector machine; compressed convex hull; kernel parameter; geometric approach; probabilistic speed-up