

基于模糊 C-均值的 相似性特征转换光滑支持向量机

方佳艳^{1,2}, 刘 峤^{1,2}, 吴 德³, 秦志光^{1,2}

(1. 电子科技大学信息与软件工程学院, 四川成都 611731; 2. 网络与数据安全省级重点实验室(电子科技大学), 四川成都 611731;
3. 西安电子科技大学计算机学院, 陕西西安 710071)

摘 要: 在用于非线性分类的光滑支持向量机(SSVM)模型中,核函数必须满足 Mercer's 条件,由此限制了核函数的选择范围;并且在面对大规模数据集时,SSVM 模型的计算复杂度很高,训练时间长. 针对这两点缺陷提出了基于模糊 C-均值的相似性特征转换光滑支持向量机模型(SFT-SSVM-FCM). 首先,运用基于相似性的特征转换,使得核函数不需要再满足 Mercer's 条件,从而拓宽了核函数的选择范围;其次,运用模糊 C-均值(FCM)分群技术,将完整的训练数据集划分成若干子簇,分别在每一个子簇上进行已经过相似性特征转换的 SSVM 模型训练. 实验表明:与传统的 SVM、SSVM 模型及一系列变体模型相比较,该新模型在训练时间、分类精度方面都具有更好的表现.

关键词: 光滑支持向量机; 模糊 C-均值; 相似性; 特征转换

中图分类号: TP181 **文献标识码:** A **文章编号:** 0372-2112 (2018)11-2714-11

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2018.11.019

Smooth Support Vector Machine with Similarity-Based Feature Transformation Technique and Fuzzy C-Means Clustering

FANG Jia-yan^{1,2}, LIU Qiao^{1,2}, WU De³, QIN Zhi-guang^{1,2}

(1. School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China;
2. Provincial Key Laboratory of Network and Data Security, Chengdu, Sichuan 611731, China;
3. School of Computer Sciences, Xidian University, Xi'an, Shaanxi 710071, China)

Abstract: We propose a new model called smooth support vector machine with similarity-based feature transformation and fuzzy C-means (FCM) clustering (SFT-SSVM-FCM). When the similarity-based feature transformation technique is applied, Mercer's conditions are no longer required for kernel functions, thus broadening the range of usable kernel functions. We also incorporate the Fuzzy-C means clustering technique to divide a whole dataset into several clusters each of which is used to perform SSVM with similarity-based feature transformation. The experimental results indicate that the proposed model has better performance compared with the conventional SVM and SSVM model as well as some variants in terms of classification accuracy and training time.

Key words: smooth support vector machine; fuzzy C-means (FCM); similarity; feature transformation

1 引言

在统计学习和优化理论中,对于小样本的分类问题,支持向量机^[1,2]是一种非常高效的分类模型,并且因其选择参数少,泛化性好以及能够得到全局最优解的特性从而被广泛使用^[3-6],同时也衍生出很多的变体

模型^[7,8]. 其中,支持向量机最显著的一个特点就是可以利用核函数解决高维空间中的向量内积的复杂运算问题. 然而,核函数的选择必须要严格满足 Mercer's 条件. 只有具备对称性,并且对应的 Gram 矩阵具备半正定的特性的函数才能作为一个有效的核函数来使用. 由此就限制了核函数的选择范围,一些性能高效并且

收稿日期:2017-10-15;修回日期:2018-02-21;责任编辑:孙瑶

基金项目:国家 863 高技术研究发展计划(No. 2011AA010706);国家自然科学基金重点项目(No. 61133016, No. 61772117);四川省高新技术及产业化面上项目(No. 2017GZ0308)

形式简单的核函数(例如 compact support kernel function)不能被用于支持向量机的模型。

为了突破这一限制,我们考虑一种类似于文献[3]中的对于原始数据特定的特征转换,并在此基础上加以改进.由于核函数本身可以作为两个数据点之间相似性的度量,本文提出了一种基于相似性的对原始数据的特征转换.我们不去计算一个数据点和其他所有数据点的相似性,而是先利用 K-均值算法从所有的训练数据中选择一些代表点,然后分别去计算一个特定的数据点与这些代表点之间的相似性,将这个特定数据点与每一个代表点之间的相似性作为该特定数据点的特征转换向量的其中一个维度,从而得到了最终的特征转换向量.和文献[3]中提出的特征转换相比,我们利用了 K-均值算法剔除了文献[3]的特征转换向量中的一些物理含义相似的特征维度,从而降低了特征转换向量的维度,提高了训练速度并且减少了预测时间。

然而,基于支持向量机(SVM)模型,我们仍然需要去解决一个有条件的最佳化问题,因此我们需要引入拉格朗日乘子将其转换成为对偶形式的二次规划问题求得全局最优解.由此,存在两点缺陷:第一,有条件的二次凸优化问题限制了无条件优化领域中的一些高效的优化算法的使用;第二,当需要处理大样本数据集时,该算法的训练时间比较长,效率低下。

Yuhje Lee 和 O L Mangarasian 提出了光滑支持向量机(SSVM)^[9],它是一种利用了平滑技术的光滑可微模型. SSVM 具有严格的凸性和无限可微的数学特性,并且引入了高效的无约束优化问题,从而吸引了很多学者从不同的角度研究 SSVM 模型^[10-16]:有的提出了不同的平滑函数^[10,12-15],有些则将其拓展到预测领域^[5,11].其他有关 SVM 和 SSVM 的应用在^[17-19]中有所介绍. LIANG 提出了一种精确的光滑模型 SDWNSVM^[16],从而收到了良好的效果,并且证明了该精确模型和原始 SSVM 模型之间的等价性.光滑支持向量机模型克服了 SVM 算法的第一个缺陷,但是并没有解决第二个缺陷. LIANG 还提出了一种隐空间光滑支持向量机模型(SHSSVM)^[20,21],但是仍然具有和 SSVM 相同的缺陷,即对于大样本数据集来说,具有较长的训练时间。

因此我们需要去建立一种训练时间较短的模型,即使是面对大规模的数据集,仍然可以具备很高的时间效率. REN 在文献[22]中指出:分段线性识别可以用来近似不同形状的超平面,最后的决策平面由多个分段线性函数的子超平面组合而成.该模型具有很强的适应性和分类能力,在线性和非线性识别上做到了一个很好的折中. REN 还将分片技术运用到了 SVM 模型当中,并取得了较好的效果,但是并没有将它延伸到

HSSVM 模型中.采用分段识别的思想构造分片支持向量机,利用样本特征进行等间距分割,导致某些子空间仅含一类样本,分类精度较差,此时需要缩减子空间的数目,并重新分割训练数据集,故而增加了训练时间.文献[10]和文献[23]运用了分段函数近似支持向量机中的松弛向量的加函数,导出光滑可微的模型,但这有悖于分段识别思想的初衷。

受隐空间^[4,6,24],平滑技术以及分段线性识别^[3,9,22]思想的启发,本文提出了一种新的模型:基于模糊 C-均值的相似性特征转换光滑支持向量机.首先,采用 K-均值算法从训练数据集中选出一些代表点,然后分别计算特定的数据点和所有代表点之间的相似性,将该数据点和每一个代表点的相似性度量作为该原始数据点的特征转换向量的一个维度.完成了特征转换之后,我们借助于模糊 C-均值聚类算法将整个训练数据集分成若干子簇,每一个子簇分别用来训练经过了基于相似性的特征转换的 SSVM 模型.最后,引入了一系列结合规则将来自于不同子簇的分类器结合形成最终的决策函数。

2 基于模糊 C-均值的相似性特征转换光滑支持向量机

给定一个用于二元分类的训练数据集: $T = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, $\mathbf{x}_i \in \mathbf{R}^n$ 是训练样本, $y_i \in \{1, -1\}$ 是训练样本对应的标签. $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ 和 $Y = \{y_i\}_{i=1}^N$ 分别是样本集合和标签集合.因此用 $T = (\mathbf{X}, Y)$ 来表示二元分类训练数据集。

SFT-SSVM-FCM 包含了四个部分:基于相似性的特征转换;模糊 C-均值聚类;在子簇中训练多个光滑支持向量机模型;对多个子簇训练产生的分类器的结合.算法的具体流程如图 1 所示。

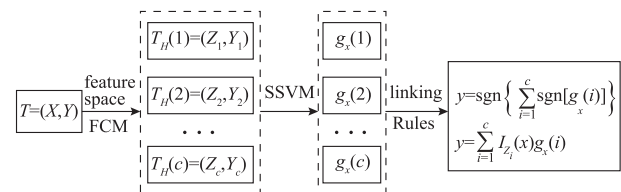


图1 SFT-SSVM-FCM算法的流程

2.1 基于相似性的特征转换

假设 $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ 是独立同分布的数据集. 定义一个实函数集合:

$$\{\varphi_i(\mathbf{x}) \mid i = 1, 2, \dots, d\} \quad (1)$$

也就是说,输入数据点被映射到了一个 d 维特征空间中,如下

$$\mathbf{x} \xrightarrow{\varphi} \mathbf{z} = [\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}), \dots, \varphi_d(\mathbf{x})]^T \quad (2)$$

特征空间 \mathbf{Z} 被定义为

$$\mathbf{Z} = \{\mathbf{z} | \mathbf{z} = [\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}), \dots, \varphi_d(\mathbf{x})]^T, \mathbf{x} \in \mathbf{X}\} \quad (3)$$

选取一个对称的核函数 $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{y}, \mathbf{x})$, 核函数映射如下

$$\mathbf{x} \xrightarrow{k} \mathbf{z} = [k(\mathbf{x}_1, \mathbf{x}), k(\mathbf{x}_2, \mathbf{x}), \dots, k(\mathbf{x}_N, \mathbf{x})] \quad (4)$$

基于对称核函数的特征转换, N 维特征空间可以表示成如下形式

$$\mathbf{Z} = \{\mathbf{z} | \mathbf{z} = [k(\mathbf{x}_1, \mathbf{x}), k(\mathbf{x}_2, \mathbf{x}), \dots, k(\mathbf{x}_N, \mathbf{x})]^T, \mathbf{x} \in \mathbf{X}\} \quad (5)$$

常用核函数如下

多项式核函数:

$$k(\mathbf{x}, \mathbf{y}) = [(\mathbf{x}^T \mathbf{y}) + 1]^q, q \in \mathbf{N} \quad (6)$$

高斯径向基函数:

$$k(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2 / \sigma^2) \quad (7)$$

S-形核函数:

$$k(\mathbf{x}, \mathbf{y}) = \tanh(p_1 \mathbf{x}^T \mathbf{y} + p_2), \quad (8)$$

$$p_1, p_2 \in \mathbf{R}$$

紧凑支持核函数:

$$k(\mathbf{x}, \mathbf{y}) = \begin{cases} \cos(p \|\mathbf{x} - \mathbf{y}\|^2), & p \|\mathbf{x} - \mathbf{y}\|^2 \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

由于这些核函数可以作为两个数据点之间相似性的度量, 因此式(5)中的每一个维度就代表了一个相似性. 对于一个给定的数据点, 该数据点和它附近的训练数据点之间的相似性度量结果是相近的, 这些相近的相似性度量在式(5)向量中可以被剔除. 因此我们运用 K-均值算法从训练数据集中选择一些代表点, 对于给定的数据点, 只和这些代表点作相似性度量, 从而降低了式(5)向量的维度, 降维后的特征向量如下

$$\mathbf{Z} = \{\mathbf{z} | \mathbf{z} = [k(\mathbf{x}_1, \mathbf{x}), k(\mathbf{x}_2, \mathbf{x}), \dots, k(\mathbf{x}_l, \mathbf{x})]^T, \mathbf{x} \in \mathbf{X}\} \quad (10)$$

通过 K-均值算法选出了 l 个代表点.

2.2 特征空间中的模糊 C-均值(FCM)集群

引理 1 在特征空间 \mathbf{H} 中, 两个数据点 \mathbf{x} 和 \mathbf{y} 之间的距离如下

$$d_H(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_H = \sqrt{\sum_{i=1}^l [k(\mathbf{x}_i, \mathbf{x}) - k(\mathbf{x}_i, \mathbf{y})]^2} \quad (11)$$

证明 在特征空间 \mathbf{H} 中, 经过了相似性特征转换的数据点 \mathbf{x} 和 \mathbf{y} 分别为

$$\mathbf{z}(\mathbf{x}) = [k(\mathbf{x}_1, \mathbf{x}), k(\mathbf{x}_2, \mathbf{x}), \dots, k(\mathbf{x}_l, \mathbf{x})]^T$$

$$\mathbf{z}(\mathbf{y}) = [k(\mathbf{x}_1, \mathbf{y}), k(\mathbf{x}_2, \mathbf{y}), \dots, k(\mathbf{x}_l, \mathbf{y})]^T$$

$k(\mathbf{x}, \mathbf{y})$ 是对称的核函数.

它们距离的平方为

$$\begin{aligned} d_H^2(\mathbf{x}, \mathbf{y}) &= \|\mathbf{x} - \mathbf{y}\|_H^2 \\ &= \|\mathbf{z}(\mathbf{x}) - \mathbf{z}(\mathbf{y})\|^2 \\ &= [k(\mathbf{x}_1, \mathbf{x}) - k(\mathbf{x}_1, \mathbf{y})]^2 \\ &\quad + [k(\mathbf{x}_2, \mathbf{x}) - k(\mathbf{x}_2, \mathbf{y})]^2 \\ &\quad + \dots + [k(\mathbf{x}_l, \mathbf{x}) - k(\mathbf{x}_l, \mathbf{y})]^2 \\ &= \sum_{i=1}^l [k(\mathbf{x}_i, \mathbf{x}) - k(\mathbf{x}_i, \mathbf{y})]^2 \end{aligned} \quad (12)$$

证毕

2.2.1 密度指数

模糊 C-均值比较容易实现, 但是它的性能表现取决于初始设定的子簇中心点. 因此, 可以考虑选择密度指数较大的样本点作为初始的子簇中心点.

在特征空间中, 数据点 \mathbf{x}_i 的密度指数被定义为

$$p_i^0 = \sum_{j=1}^l \exp(-\alpha \|\mathbf{x}_i - \mathbf{x}_j\|_H^2) \left(\alpha = \frac{4}{\gamma_a^2} \right) \quad (13)$$

半径 r_a 是一个正数. 所有在该半径以外的数据都对密度指数的计算影响比较小. 显然, 当 \mathbf{x}_i 周围聚集的数据越多, \mathbf{x}_i 的密度指数就越大. 由于密度指数的计算是一种指数运算, 当数据量很大时, 计算速度比较慢, 通常在实际上采用如下公式来计算 \mathbf{x}_i 的密度指数.

$$p_i^0 = \sum_{j=1}^l \frac{1}{1 + \alpha \|\mathbf{x}_i - \mathbf{x}_j\|_H^2} \left(\alpha = \frac{4}{\gamma_a^2} \right) \quad (14)$$

取

$$P_i^* = \max \{P_i^0; i = 1, 2, \dots, l\} \quad (15)$$

然后选择对应的 \mathbf{x}_i^* 作为第一个初始的子簇中心点. 密度指数按照如下公式作更新

$$p_i^{(k)} = p_i^{(k-1)} - p_i^* \exp\left[-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_H^2}{(\gamma_a/2)}\right] \quad (16)$$

P_k^* 的计算如下

$$P_k^* = \max \{P_i^{(k-1)}; i = 1, 2, \dots, l\} \quad (17)$$

\mathbf{x}_k^* 是被选择出来的第 k 个初始子簇中心点.

上述的 γ_a 不是一个常数, 而是与产生样本的分布的特性相关. γ_a 的计算公式如下

$$\gamma_a = \frac{1}{2} \sqrt{\frac{1}{n(n-1)} \sum_{k=1}^c \sum_{i=1}^l \|\mathbf{x}_i - \mathbf{x}_k\|_H^2} \quad (18)$$

2.2.2 模糊 C-均值

模糊 C-均值算法用于将已经处于特征空间中的样本集分割成 C 个子簇 $\{Z_j\}_{j=1}^C$, 满足 $Z = \bigcup_{j=1}^C Z_j$ 和 $Z_i \cap Z_j = \emptyset (i \neq j)$. 模糊 C-均值的实现实际上是解决如下有条件的优化问题

$$\begin{aligned} \min_{(U, V)} J_m(U, V, Z) &= \sum_{i=1}^c \sum_{j=1}^l (u_{ij})^m d_{ij}^2 \\ \text{s. t. } \sum_{i=1}^c u_{ij} &= 1, \forall j = 1, 2, \dots, l \end{aligned} \quad (19)$$

J_m 是损失函数, $\mathbf{V} = (v_1, v_2, \dots, v_c)$ 是子簇中心点, $\mathbf{U} = (u_{ij})_{c \times l}$ 是隶属度矩阵, 其中 $u_{ij} \in [0, 1]$ 描述了样本 \mathbf{x}_j

对应于中心点 v_i 的隶属度. $m \geq 1$ 是一个加权指数,用来控制集群的模糊程度. 模糊性和 m 成正相关. 根据引理 1, d_{ij} 是 v_i 和 x_j 在特征空间中的距离.

$$d_H(i, j) = \|x_i - v_j\|_H = \sqrt{\sum_{i=1}^l [k(x_i, x_j) - k(x_i, x_i)]^2} \quad (20)$$

对于特征空间中的样本集的最优的划分就是找到 J_m 的局部最优解. 在 FCM 算法中, 最小化 J_m 是基于交叉优化的方法, 根据以下两个公式分别对中心点集合和隶属度集合作交叉优化.

$$v_i^{(r)} = \frac{\sum_{j=1}^n (u_{ij}^{(r)})^m x_j}{\sum_{j=1}^n (u_{ij}^{(r)})^m} \quad (i=1, 2, \dots, c) \quad (21)$$

$$u_{ij}^{(r+1)} = \frac{1}{\sum_{k=1}^c \left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{2}{m-1}}} \quad (22)$$

通过对 J_m 的偏导数为零得到式(21)和式(22).

为了确定子簇的数量, 采用划分系数子簇有效函数(partition coefficient cluster validity function)

$$F = \frac{\sum_{i=1}^c \sum_{j=1}^l (u_{ij})^m}{l} \quad (23)$$

2.3 特征空间中的光滑模型

2.3.1 光滑支持向量机

$\xi = [\xi_1, \xi_2, \dots, \xi_N]^T$ 是有松弛变量组成的松弛向量, $C > 0$ 是惩罚系数, 可以在最大边界和边界违反量之间做出一个折中. 训练光滑支持向量机模型等价于解决一个有约束的二次凸优化问题

$$\min \frac{1}{2}(\mathbf{w}^T \mathbf{w} + b^2) + C \sum_{i=1}^l \xi_i^2 \quad (24)$$

$$\text{s. t. } y_i(\mathbf{w}^T \mathbf{z}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0, i=1, \dots, N$$

其中, $\mathbf{w} \in \mathbf{R}^l$ 是权值向量, $b \in \mathbf{R}^1$ 是控制超平面相对于原点的偏移量.

用熵函数:

$$P_\beta(\mathbf{x}) = \mathbf{x} + \beta^{-1} \ln[1 + \exp(-\beta \mathbf{x})] \quad (25)$$

去近似松弛向量的加函数:

$$\xi = \max[\mathbf{e} - \mathbf{D}(\mathbf{A}\mathbf{w} + b\mathbf{e}), \mathbf{0}] = [\mathbf{e} - \mathbf{D}(\mathbf{A}\mathbf{w} + b\mathbf{e})]_+ \quad (26)$$

其中, 记 $\mathbf{A} = [\mathbf{z}_1^T, \mathbf{z}_2^T, \dots, \mathbf{z}_N^T]^T$ 是训练数据集的矩阵形式, \mathbf{e} 是由 1 组成的列向量, \mathbf{D} 是一个对角矩阵, 其对角线元素是 +1 或 -1, 对应于 x_i 的标签值. 由此, 对于光滑支持向量机(SSVM)的训练如下

$$\min_{\mathbf{w}, b} \frac{1}{2}(\mathbf{w}^T \mathbf{w} + b^2) + C \|\mathbf{P}_\beta[\mathbf{e} - \mathbf{D}(\mathbf{A}\mathbf{w} + b\mathbf{e})]\|^2 \quad (27)$$

这是一个无约束的优化问题, 因此可以使用 Newton-Armijo 算法求得全局最优解. 详细步骤参考文献[9].

2.3.2 基于模糊 C-均值的光滑支持向量机模型

在特征空间里使用模糊 C-均值聚类算法将训练样本划分成 c 个子簇. 为方便直觉上的理解, 将第 i 个子簇的样本记为 $\mathbf{z}_k(i)$, 对应的标签值为 $y_k(i) \in \{-1, 1\}$. 将 $(\mathbf{z}_k(i), y_k(i))$ 按照下标 k 的大小升序排列, 得到第 i 个子簇的训练数据集

$$T_H(i) = (Z_i, Y_i) = \{(\mathbf{z}_k(i), y_k(i))\}_{k=i_1}^{i_2} \quad (28)$$

其中 $Z_i = \{\mathbf{z}_k(i)\}_{k=i_1}^{i_2}$, $Y_i = \{y_k(i)\}_{k=i_1}^{i_2}$ ($i_1 < i_2 < \dots < i_s$), 第 i 个子簇可以表示为

$$Z_i = \left\{ \mathbf{z}_k \mid \mathbf{z}_k = \begin{bmatrix} k(\mathbf{x}_i, \mathbf{x})(i), k(\mathbf{x}_{i_2}, \mathbf{x})(i), \\ \dots, k(\mathbf{x}_{i_s}, \mathbf{x})(i) \end{bmatrix}^T, \mathbf{x} \in X_i \right\} \quad (29)$$

其中 $X_i = \{\mathbf{x}_k(i)\}_{k=i_1}^{i_2}$, $k_{(x_p, x_q)}(i) = \varphi(x_p)^T \varphi(x_q)$ 是核函数. 记 $\xi(i) = \{\xi_k(i)\}_{k=i_1}^{i_2}$ ($\xi_k(i) \geq 0$) 为子簇 $X(i)$ 的松弛向量, 记惩罚常数为 $C(i) > 0$ 来权衡边界最大化和边界违反量. 第 i 个子簇对应的光滑支持向量机模型如下

$$\min_{\mathbf{w}, b} \frac{1}{2}[\mathbf{w}(i)^T \mathbf{w}(i) + b(i)^2] + C(i) \|\mathbf{P}_\beta\{\mathbf{e}(i) - \mathbf{D}(i)[\mathbf{A}(i)\mathbf{w}(i) + b(i)\mathbf{e}(i)]\}\|^2 \quad (30)$$

梯度向量为

$$\nabla F_\beta \begin{pmatrix} \mathbf{w}(i) \\ b(i) \end{pmatrix} = \frac{1}{2C(i)} \begin{bmatrix} \mathbf{w}(i) \\ b(i) \end{bmatrix} + \begin{bmatrix} -\mathbf{Q}(i)^T \\ -\mathbf{Y}(i)^T \end{bmatrix} \cdot \text{Diag}\{\mathbf{P}_\beta[\mathbf{t}(i)]\} \cdot \text{Diag}[\mathbf{e}(i) + \mathbf{v}(i)^{-1}]\mathbf{e}(i) \quad (31)$$

Hessian 矩阵为

$$\nabla^2 F_\beta \begin{pmatrix} \mathbf{w}(i) \\ b(i) \end{pmatrix} = \frac{1}{2C(i)} \begin{bmatrix} \mathbf{I}(i) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} + 2 \begin{bmatrix} -\mathbf{Q}^T(i) \\ -\mathbf{Y}^T(i) \end{bmatrix} \cdot \text{Diag}[\mathbf{e}(i) + \mathbf{v}(i)]^{-2} \mathbf{M}[-\mathbf{Q}(i) - \mathbf{Y}(i)] \quad (32)$$

其中: $\mathbf{Q}(i) = \mathbf{D}(i)\mathbf{A}(i)$

$$\mathbf{Y}(i) = \mathbf{D}(i)\mathbf{e}(i)$$

$$\mathbf{t}(i) = \mathbf{e}(i) - \mathbf{D}(i)[\mathbf{A}(i)\mathbf{w}(i) - \mathbf{e}(i)b(i)]$$

$$\mathbf{v}(i) = \exp[-\beta \cdot \mathbf{t}(i)]$$

$$\mathbf{M} = \{\mathbf{I}(i) + \beta \cdot \text{Diag}[\mathbf{v}(i)]\} \cdot \text{Diag}\{\mathbf{P}_\beta[\mathbf{t}(i)]\}$$

显然, Hessian 矩阵是半正定的, 所以该无约束优化问题属于凸优化问题, 存在全局最优解. 因此, 像一些高效的无约束优化算法如 Newton-Armijo 算法, 共轭梯度下降算法都可以用来得到光滑支持向量机模型的全局最优解. 同时可以推导出算法收敛的结果, 推导过程类似于文献[8]中的定理 2.

定理 1 记 $\mathbf{A}(i) = [\mathbf{z}_{i_1}^T, \mathbf{z}_{i_2}^T, \dots, \mathbf{z}_{i_i}^T]^T, b(i) \in \mathbf{R}^1$. 定义了二个实函数 $F\left(\frac{\mathbf{w}(i)}{b(i)}\right), F_\beta\left(\frac{\mathbf{w}(i)}{b(i)}\right)$, 它们对应的优化问题如下:

$$\min_{\mathbf{w}(i), b(i)} F\left(\frac{\mathbf{w}(i)}{b(i)}\right) = \frac{1}{2}[\mathbf{w}(i)^T \mathbf{w}(i) + b(i)^2] + C(i) \cdot \|\{\mathbf{e}(i) - \mathbf{D}(i)[\mathbf{A}(i)\mathbf{w}(i) + b(i)\mathbf{e}(i)]\}_+\|^2 \quad (33)$$

$$\min_{\mathbf{w}(i), b(i)} F_\beta\left(\frac{\mathbf{w}(i)}{b(i)}\right) = \frac{1}{2}[\mathbf{w}(i)^T \mathbf{w}(i) + b(i)^2] + C(i) \cdot \|P_\beta\{\mathbf{e}(i) - \mathbf{D}(i)[\mathbf{A}(i)\mathbf{w}(i) + b(i)\mathbf{e}(i)]\}\|^2 \quad (34)$$

其中, $\beta > 0$ 是平滑参数, 则以下三个成立.

(1) 这两个优化问题都具有全局最优解.

(2) 记 $\left(\frac{\bar{\mathbf{w}}(i)}{\bar{b}(i)}\right)$ 和 $\left(\frac{\bar{\mathbf{w}}_\beta(i)}{\bar{b}_\beta(i)}\right)$ 分别为 $\min F\left(\frac{\mathbf{w}(i)}{b(i)}\right)$ 和 $\min F_\beta\left(\frac{\mathbf{w}(i)}{b(i)}\right)$ 这两个无约束优化问题的全局最小解, 则下面的不等式成立

$$\|\bar{\mathbf{w}}_\beta(i) - \bar{\mathbf{w}}(i)\|^2 + \|\bar{b}_\beta(i) - \bar{b}(i)\|^2 \leq s \left[\left(\frac{\ln 2}{\beta}\right)^2 + \frac{2\rho \ln 2}{\beta} \right] \quad (35)$$

其中 ρ 被定义为

$$\rho = \max_{k=i_1, i_2, \dots, i_i} \|\{\mathbf{e}(i) - \mathbf{D}(i)[\mathbf{A}(i)\mathbf{w}(i) + b(i)\mathbf{e}(i)]\}_k\| \quad (36)$$

(3) 当平滑参数 β 趋近于无限大时, 近似解 $\left(\frac{\bar{\mathbf{w}}_\beta(i)}{\bar{b}_\beta(i)}\right)$ 收敛于真实解 $\left(\frac{\bar{\mathbf{w}}(i)}{\bar{b}(i)}\right)$.

我们采用 Newton-method 最优化算法得到最优解 $\mathbf{w}(i)^*$ 和 $b(i)^*$. 更多细节请参考文献[9]. 对于第 i 个子簇 Z_i 和未知的测试样本 \mathbf{x} , 用于分类的超平面和决策函数如下

$$g_x(i) = \mathbf{w}(i)^T \mathbf{z}(\mathbf{x}) + b(i) = 0 \quad (37)$$

$$y_x(i) = \text{sgn}(g_x(i)) = \text{sgn}\{\mathbf{w}(i)^* \mathbf{z}(\mathbf{x}) + b(i)^*\} \quad (38)$$

其中, $\mathbf{z}(\mathbf{x})$ 是数据 \mathbf{x} 在特征空间中的转换形式

$$\mathbf{Z} = \{\mathbf{z} | \mathbf{z} = [k(\mathbf{x}_1, \mathbf{x}), k(\mathbf{x}_2, \mathbf{x}), \dots, k(\mathbf{x}_i, \mathbf{x})]^T, \mathbf{x} \in \mathbf{X}\} \quad (39)$$

2.4 结合规则

每一个完整的训练过程都是建立在一个子簇上的, 因此多个基于相似性特征转换的 SSVM 模型可以在这些子簇上并行训练, 比起传统的 SSVM 模型, 这样做极大地减少了训练时间. 在测试阶段同样可以并行实现多个 SSVM 模型, 然后按照某种结合规则, 将来自多个子簇的分类器结合起来形成最终的决策函数.

2.4.1 线性结合 (weighting rule)

我们可以考虑对来自不同子簇的多个分类器进行线性结合. 对于一个测试数据 \mathbf{x} , 每一个子簇的分类器对最终的标签预测都会有所贡献, 贡献的大小和该分类器对应的子簇的规模成正比. 那些来自规模较小的子簇的分类器在计算标签值时就会产生比较小的影响. 这样的线性结合可以让我们得到更稳定的预测结果. 最终的决策函数如下

$$y = \text{sgn} \sum_{i=1}^c \alpha(i) y_x(i) = \text{sgn} \left\{ \sum_{i=1}^c \alpha(i) \text{sgn}[g_x(i)] \right\} \quad (40)$$

其中, $y_x(i) = \text{sgn}(g_x(i))$ 是来自第 i 个子簇 Z_i 的分类器的预测结果.

2.4.2 吸收规则 (absorbing rule)

首先, 我们需要确定测试样本处在哪一个子簇当中, 之后再决定决策函数. 也就是说, 我们只使用该测试样本所在子簇的分类器来组成最后的决策函数, 而忽略来自其他子簇的分类器. 决策函数的构造如下

$$y = \sum_{i=1}^c I_{Z_i}(\mathbf{x}) g_x(i) \quad (41)$$

其中 $g_x(i)$ 是第 i 个子簇对应的最优决策边界, $I_{Z_i}(\mathbf{x})$ 是示性函数.

$$I_{Z_i}(\mathbf{x}) = \begin{cases} 1, & \mathbf{z}(\mathbf{x}) \in Z_i \\ 0, & \mathbf{z}(\mathbf{x}) \notin Z_i \end{cases} \quad (42)$$

为了确定测试样本 \mathbf{x} 是属于哪一个子簇, 在特征空间中根据不同的用户定义的距离提出了以下三种在特征空间下度量距离的方法.

方法一 在特征空间中, 数据点 \mathbf{x} 和子簇中心点 \mathbf{v}_i ($i = 1, 2, \dots, c$) 之间的距离被定义为

$$d_{ij} = \|\mathbf{x} - \mathbf{v}_i\|_H = \sqrt{\sum_{t=1}^i [k(\mathbf{x}_t, \mathbf{x}) - k(\mathbf{x}_t, \mathbf{v}_i)]^2} \quad (43)$$

方法二 在特征空间中, 数据点 \mathbf{x} 和子簇中心点 \mathbf{v}_i ($i = 1, 2, \dots, c$) 之间的适应性距离被定义为

$$d_i = \frac{\|\mathbf{x} - \mathbf{v}_i\|_H}{r_i} = \frac{\sqrt{\sum_{t=1}^i [k(\mathbf{x}_t, \mathbf{x}) - k(\mathbf{x}_t, \mathbf{v}_i)]^2}}{r_i} \quad (44)$$

其中, r_i 是第 i 个子簇的半径, 被定义为

$$r_i = \frac{\sum_{n=1}^N \|\mathbf{x}_n - \mathbf{v}_i\|_H}{\text{mod } X_i} = \frac{\sum_{n=1}^N \sqrt{\sum_{t=1}^i [k(\mathbf{x}_t, \mathbf{x}_n) - k(\mathbf{x}_t, \mathbf{v}_i)]^2}}{\text{mod } X_i} \quad (45)$$

其中, $\mathbf{x}_n \in X_i$ 且 $\text{mod } X_i$ 表示第 i 个子簇所包含的样本数

量.

方法三 在特征空间中,数据点 x 到第 i 个子簇的最优分类超平面的距离为

$$d_i = \frac{[\mathbf{w}(i)^*]^T \mathbf{z}(x) + b(i)^*}{\|\mathbf{w}(i)^*\|} \quad (46)$$

其中, $\mathbf{w}(i)^*$ 和 $b(i)^*$ 分别是第 i 个子簇的最优分类超平面的权值向量和偏移量. 找出 d_i 的最小值并存储下标 $j(j = \arg \min(d_i))$, 然后用该下标对应的子簇的分类器为测试数据点赋标签值, 采用的决策函数如式(38).

3 SFT-SSVM-FCM 模型的实现及分析

我们所提出的新模型首先利用 K-均值算法选出整个训练数据集的代表点, 并实现对原始数据集的相似性特征转换. 然后, 在转换之后的特征空间中, 采用模糊 C-均值集群技术将整个经过了特征转换的数据集划分成若干子簇, 根据每一个数据点的密度指数的大小选择初始的子簇中心点. 之后在每一个子簇上并行训练基于相似性特征转换的 SSVM 模型. 最后使用结合规则组成最终的决策函数.

3.1 实现

第一部分:相似性特征转换

我们首先在整个原始数据集上训练 K-均值算法. 首先随机选择出 l 个原始数据点分别作为初始的 c 个子簇的中心点. 然后分别针对 $\{\boldsymbol{\mu}_m\}$ 和 $\{S_m\}$ 对式(47)作最小化, 其中 $\boldsymbol{\mu}_m$ 是第 m 个子簇的中心点, S_m 是组成第 m 个子簇的所有原始数据点的集合.

$$E(S_1 \cdots S_l; \boldsymbol{\mu}_1 \cdots \boldsymbol{\mu}_l) = \frac{1}{N} \sum_{n=1}^N \sum_{m=1}^l [\mathbf{x}_n \in S_m] \|\mathbf{x}_n - \boldsymbol{\mu}_m\|^2 \quad (47)$$

$$\boldsymbol{\mu}_m = \frac{\sum_{\mathbf{x}_n \in S_m} \mathbf{x}_n}{|S_m|} \quad (48)$$

算法收敛时, 将每一个原始数据点和这些中心点分别做相似性计算, 完成相似性特征转换(5).

第二部分:模糊 C-均值集群算法

第 1 步:初始化

输入已经完成特征转换的训练数据集 $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N\}$, 子簇数量 c , 阈值 ε_1 , 初始子簇中心点 $\mathbf{V} = (\mathbf{z}_1^*, \mathbf{z}_2^*, \dots, \mathbf{z}_c^*)$, 隶属度矩阵 $\mathbf{U}^{(r)} = (u_{ij}^{(r)})_{N \times c}$, 其中隶属度矩阵中的元素值从 $[0, 1]$ 区间中随机取值作为初始值, FCM 中的权重指数 $m = 2$, 迭代次数 $r = 0$.

第 2 步:迭代

用式(18)和式(19)计算并更新子簇中心点 $v_i^{(r)}$ 和隶属度矩阵元素 $u_{ij}^{(r+1)}$, 并设置 $r := r + 1$.

第 3 步:判断算法收敛

计算损失函数 J_m^r . 如果前后两次迭代过程中的损

失函数变化量小于事先设定的阈值 ε_1 , 即 $\|J_m^r - J_m^{r-1}\| < \varepsilon_1$, 则终止 FCM 算法并输出隶属度矩阵 \mathbf{U} 以及子簇中心点 $\mathbf{V} = (\mathbf{v}_1^*, \mathbf{v}_2^*, \dots, \mathbf{v}_c^*)$, 否则设置 $r = r + 1$ 并返回第二步.

第三部分:SSVM 模型训练

让 β 趋近于无穷大, 用式(31)和式(32)的极限值去计算梯度向量和 Hessian 矩阵.

第 1 步:初始化

输入第 i 个子簇的训练数据集 $T_i = \{(\mathbf{z}_k(i), y_k(i))\}_{k=i_1}^{i_2} (i_1 < i_2 < \dots < i_s)$, 初始点为 $\begin{pmatrix} \mathbf{w}(i)^0 \\ b(i)^0 \end{pmatrix} \in \mathbf{R}^{l+1}$, 设定阈值 $\varepsilon_2 > 0$, 并计算梯度向量 ∇F_β $\begin{pmatrix} \mathbf{w}(i)^0 \\ b(i)^0 \end{pmatrix}$ 和 Hessian 矩阵 $\nabla^2 F_\beta \begin{pmatrix} \mathbf{w}(i)^0 \\ b(i)^0 \end{pmatrix}$ 的值. 设置迭代次数 $t = 0$.

第 2 步:判断终止条件

如果梯度向量在 $\begin{pmatrix} \mathbf{w}(i)^t \\ b(i)^t \end{pmatrix}$ 处的值为 0, 或者梯度向量在 $\begin{pmatrix} \mathbf{w}(i)^t \\ b(i)^t \end{pmatrix}$ 处的值小于预先设定的阈值 ε_2 , 即 $\|\nabla F_\beta \begin{pmatrix} \mathbf{w}(i)^t \\ b(i)^t \end{pmatrix}\| < \varepsilon_2$, 则算法收敛并输出分割超平面: $g_i(\mathbf{x}) = \mathbf{w}(i)^T \mathbf{z}(\mathbf{x}) + b(i)$, 否则进行后续步骤.

第 3 步:牛顿方向

通过设置 $F_\beta \begin{pmatrix} \mathbf{w}(i)^t \\ b(i)^t \end{pmatrix}$ 在 $\begin{pmatrix} \mathbf{w}(i)^t \\ b(i)^t \end{pmatrix}$ 处的线性化式子为 0, 得到下降方向 $d^t \in \mathbf{R}^{l+1}$, 从而得到式(49)

$$\nabla^2 F_\beta \begin{pmatrix} \mathbf{w}(i)^t \\ b(i)^t \end{pmatrix} d^t = -\nabla F_\beta \begin{pmatrix} \mathbf{w}(i)^t \\ b(i)^t \end{pmatrix} \quad (49)$$

第 4 步:Armijo 步长

使用 Armijo 步长确定搜索步长 λ_t , $\lambda_t = \max\{1, \frac{1}{2}, \frac{1}{4}, \dots\}$, 使得对于 $\delta \in (0, \frac{1}{2})$,

$$F_\beta \begin{pmatrix} \mathbf{w}(i)^{t+1} \\ b(i)^{t+1} \end{pmatrix} - F_\beta \begin{pmatrix} \mathbf{w}(i)^t \\ b(i)^t \end{pmatrix} + \lambda_t d^t \geq -\delta \lambda_t \nabla F_\beta \begin{pmatrix} \mathbf{w}(i)^t \\ b(i)^t \end{pmatrix} d^t \quad (50)$$

第 5 步:更新梯度方向

$$\begin{pmatrix} \mathbf{w}(i)^{t+1} \\ b(i)^{t+1} \end{pmatrix} = \begin{pmatrix} \mathbf{w}(i)^t \\ b(i)^t \end{pmatrix} + \lambda_t d^t \nabla F_\beta \begin{pmatrix} \mathbf{w}(i)^{t+1} \\ b(i)^{t+1} \end{pmatrix}$$

并计算更新之后的梯度向量的值, 然后设置 $t = t + 1$, 并返回第 2 步.

第四部分:结合规则

输入测试数据 \mathbf{x} , 并作相似性特征转换, 获得如式(37)所示的所有子簇的最优超平面. 使用线性结合规

则或吸收规则来计算测试数据的标签值。

3.2 复杂度分析

对于含有 N 个样本的训练数据集, SVM、SSVM、HSSVM 和 SDWNSVM 都是一次使用完整的训练数据集进行训练, 它们的空间复杂度为 $O(N^2)$ 。基于模糊 C-均值的相似性特征转换的光滑支持向量机模型(SFT-SSVM-FCM)利用 FCM 技术将转换之后的数据集划分成 c 个区域, 每个区域上单独训练 SSVM 模型得到最优超平面, 最后将所有的超平面进行有机整合形成最终的决策函数。假设每个子簇上面的样本点有 N/C 。我们提出的模型总的空间复杂度为 $O\left(\frac{N}{C^2}\right)$ 。由于 $O\left(\frac{N}{C^2}\right) \ll O(N^2)$, 该新模型相比于以上现有的模型具有较低的空间复杂度。由于多个子簇上的 SSVM 模型可以并行训练, 该新模型具有较短的训练时间, 因此很适合大规模样本数据集。

4 实验分析与比较

本文分别在人工和标准数据集上对提出的新模型进行验证。程序采用 Matlab 2010a 编制, 实验在内存 4G, CPU 为 i53270 的计算机上进行。

4.1 算法性能随子簇数量的变化

从正态分布中人工产生 300 个用于二元分类的数据点, 标签值为 +1 和 -1 的数据量相等。

为了便于比较, 不同算法的参数设置相同, 惩罚常数 $C = 1$, 采用高斯径向基函数 $k(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2 / \sigma^2)$, 核宽度设为 $\delta = 0.1$ 。对于不同子簇产生的多个分类器, 采用线性组合方式得到最终的决策函数。针对不同的子簇数量, 表 1 反应了新模型的算法性能随着子簇数量的变化, 并和 SVM、SSVM、HSSVM 三种算法作了对比。时间和精度分别代表在完整数据集上的训练时间和训练精度。子簇数量 $c = 1$ 代表在训练 SSVM 模型之前并没有做 FCM 集群。

表 1 不同算法的分类表现

Algorithm	Accuracies	Time
SVM	94.76%	6.51s
SSVM	94.53%	3.18s
HSSVM	93.91%	7.23s
SFT-SSVM-FCM	$c = 1$	94.72%
	$c = 2$	94.96%
	$c = 5$	95.69%
	$c = 8$	94.12%

从表 1 的数据中, 我们可以得到如下结论。

(1) 本文提出的新模型 SFT-SSVM-FCM 的分类精度都高于 94%, 并且不随子簇数量的变化而发生显著变化。

(2) SFT-SSVM-FCM 具有最短的训练时间(随着 c 值的增大而减小), 且不随子簇数量的改变而发生明显变化。这表明, 使用平滑技术求解最优化问题比传统 SVM 模型中求解对偶问题要更加高效。

实验过程中, 我们也可以使用式(23)在(1, 10)范围内寻找最优的 c 值, 最优结果为 $c = 5$, 刚好是表 1 中最高训练精度所对应的 c 值。因此, 划分系数的有效性函数(23)在后续实验中被用来选择最优的子簇数量。

4.2 算法性能随不同结合规则的变化

我们使用了三个中等规模的数据集, 数据集的特性如表 2 所示。

表 2 各个数据集的特性

Data	Training	Testing	Class	Attributes
Ionosphere	210	141	2	34
Breast Cancer	420	279	2	9
Pima Indians	460	308	2	8

对于所有的数据集, 惩罚参数 $C = 10$, 并使用了高斯径向基函数($\sigma = 0.5$)和对称的 S-形核函数($p_1 = 2, p_2 = 0.8$)。实验进行了 10 次随机抽样。由于将多个分类器结合所需要的时间和在子簇上训练出最优分类器所花费的时间比较起来可以被忽略, 所以训练时间和测试时间几乎不随结合规则的变化而变化。测试精度反应了算法的泛化能力。表 3 显示了在训练集和测试集上的平均分类精度。“Ker”是“kernel function”的缩写, “Gau”代表“Gaussian kernel”, “Sig”代表“sigmoid kernel”。

表 3 不同结合规则的分类精度

Data	Ker	Averaged Accuracy			
		Weighting Rule	Absorbing Rule 1	Absorbing Rule 2	Absorbing Rule 3
Ionosp-here	Gau	90.87%	78.61%	84.09%	91.77%
	Sig	89.93%	77.53%	85.41%	90.09%
Breast Cancer	Gau	95.14%	85.93%	93.20%	96.33%
	Sig	93.27%	81.25%	92.84%	95.18%
Pima Indians	Gau	77.08%	66.16%	74.77%	77.65%
	Sig	77.13%	62.87%	75.49%	77.30%

显然, 使用吸收规则 3(Absorbing Rule 3)具有最高的分类精度, 线性结合规则的分类精度要略低一些。但是它们的分类精度要大大高于吸收规则 1, 2 对应的分类精度。类似的结论同样适用于其他的核函数。具体分析在此省略。

4.3 算法性能随核宽度参数值的变化

Breast Cancer 数据集由 458 个“benign”样本和

241 个“malignant”样本组成,每一个样本具有 9 个属性值.对于不同的算法,设 $C = 1$.在相同核宽度参数下,将 SFT-SSVM-FCM 模型的分类精度与 SVM、SSVM 和 HSSVM 模型进行比较.随机抽取 50% 的数据作为训练数据集,其他的作为测试数据集.表 4 反应了基于 10 次随机抽样的不同算法在不同核宽度参数下的分类精度.

表 4 不同核宽度参数值的分类精度

Kernel width	Accuracy				
	SVM	SSVM	HSSVM	SFT-SSVM-FCM	
				Weighting rule	Absorbing rule 3
$\sigma = 0.01$	66.76%	63.59%	60.37%	91.42%	93.26%
$\sigma = 0.2$	89.93%	86.33%	84.29%	92.17%	93.75%
$\sigma = 0.5$	93.22%	93.20%	93.09%	95.08%	96.33%
$\sigma = 0.8$	91.08%	91.71%	90.30%	95.29%	96.37%
$\sigma = 1$	89.81%	91.06%	85.73%	93.69%	93.96%

基于表 4,我们可以得出以下结论.

(1) SFT-SSVM-FCM 具有最高的分类精度.

(2) SFT-SSVM-FCM 模型的性能对核宽度参数值不敏感,而 SVM、SVM 和 HSSVM 算法的性能对于不同的核宽度参数值会产生较为明显的变化.

当 $\sigma = 0.01$ 和 $\sigma = 0.5$ 时,采用线性结合规则时, SFT-SSVM-FCM 模型的分类精度分别为 91.42% 和 95.08%;当采用吸收规则 3 时,分类精度分别为 93.26% 和 96.33%.将核宽度值从 0.5 增加到 1 时, SFT-SSVM-FCM 的分类精度先增加后减小,但是变化范围不超过 1.6%.

当 $\sigma = 0.01$, $\sigma = 0.5$ 和 $\sigma = 1$ 时, SVM 的分类精度分别为 66.76%、93.22% 和 89.81%.当核宽度值从 0.01 增加到 0.5 时, SVM 模型的分类精度增加了 26.46%;当核宽度值从 0.5 增加到 1 时,分类精度增加了 3.41%.由此可以看出,对于 SVM 模型,分类精度随核宽度值的变化是比较明显的.

对于 SSVM 和 HSSVM 模型而言,结果是类似的.换句话说,在一定范围内,分类精度和核宽度参数值的大小是成正比的,当核宽度值增大到某特定值时,分类精度就会逐渐减小.

为了更加直观和具体,我们也使用了大规模的图像数据集去进一步阐述不同算法的性能随核宽度参数值的变化.

图像数据集由 1300 个训练样本和 1010 个测试样本组成,每一个样本由 18 个特征维度组成,惩罚参数 C

$= 1$,采用高斯核函数.不同算法的分类精度的随核宽度参数值的平均变化量趋势如图 2 所示.

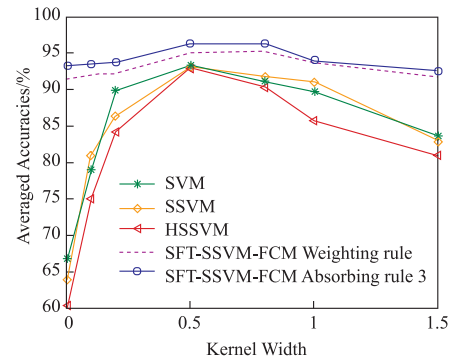


图2 分类精度随核宽度参数值的变化

显然, SFT-SSVM-FCM 具有最强的鲁棒性和最高的分类精度.并且当我们使用吸收规则 3 (Absorbing rule 3) 时,对应的精度会更高.

4.4 中等规模数据集上的表现

我们使用在四个中等规模的数据集上的训练结果来进一步阐述 SFT-SSVM-FCM 算法相比于其他算法的优越性.表 5 给出了四个中等数据集的特性.对于表中多类别的数据,第一个类别作为正类别,其他类别作为负类别.

表 5 各类数据集特性

Data	Training	Testing	Class	Attributes
Banana	400	4900	2	2
Waveform	400	4600	2	20
Iris	50	100	3	4

我们将 SFT-SSVM-FCM 模型性能和 SVM、HSSVM、SHSSVM 模型性能作了相关的比较,如表 6 所示.相关的参数设置如下: $C = 1$, $\delta = 0.5$, $p = 0.1$.“CS”是“compact support kernel function”的缩写.“RBF”代表了“Gaussian radial basis kernel function”.精度指的是训练数据集和测试数据集的平均精度.表格的空白部分表示由于内存溢出而无法计算.

从表 6 中我们可以得到如下结论.

(1) SFT-SSVM-FCM 模型具有最高的分类精度.以 Waveform 数据集为例:当采用对称且半正定的高斯核函数时, SFT-SSVM-FCM 的分类精度比 SSVM、HSSVM 和 SHSSVM 的分类精度分别高出 3.42%、2.76% 和 3.18%;当采用对称但不是半正定的紧凑支持核函数 (compact support kernel) 时, SFT-SSVM-FCM 的分类精度比 HSSVM 和 SHSSVM 的分类精度分别高出 2% 和 3.6%.由此说明采用 FCM 技术将整个训练数据集分群,然后将来自每一个子簇的分类器线性组合在一起可以获得良好的泛化能力.

表 6 不同算法在各个数据集上的比较

Data	Algorithm	Ker	Accuracies	Time	Iteration	
Banana	SSVM	RBF	89.67%	2.59s	3.1	
		CS	\	\	\	
	HSSVM	RBF	88.82%	14.66s	\	
		CS	90.71%	16.18s	\	
	SHSSVM	RBF	89.81%	2.93s	3.2	
		CS	90.92%	3.01s	3.1	
	SFT-SSVM-FCM	RBF	92.18%	1.21s	1.9	
		CS	93.33%	1.16s	2.0	
	Wave-form	SSVM	RBF	89.93%	22.36s	5.1
			CS	\	\	\
HSSVM		RBF	90.59%	63.71s	\	
		CS	92.01%	74.53s	\	
SHSSVM		RBF	90.17%	21.24s	5.3	
		CS	90.41%	21.97s	4.9	
SFT-SSVM-FCM		RBF	93.35%	10.03s	3.8	
		CS	94.01%	8.96s	4.0	
Iris		SSVM	RBF	96.67%	0.46s	2.9
			CS	\	\	\
	HSSVM	RBF	95.47%	2.62s	\	
		CS	95.55%	2.67s	\	
	SHSSVM	RBF	96.81%	0.46s	3.2	
		CS	97.13%	0.46s	2.9	
	SFT-SSVM-FCM	RBF	99.73%	0.21s	1.1	
		CS	99.96%	0.17s	1.3	

(2) 当采用对称或者半正定的核函数时, SFT-SSVM-FCM 具有最短的训练时间. 在 Waveform 数据集上, 当采用半正定的“RBF”核函数时, SFT-SSVM-FCM 的训练时间是 HSSVM 模型的训练时间的 15.74%. 同时, 该新模型的训练时间分别是 SSVM 和 SHSSVM 模型训练时间的 44.86% 和 47.22%. 这就验证了使用 FCM 技术将整个经过特征转换的训练数据集分成若干子簇的方法确实能够降低训练复杂度.

(3) SFT-SSVM-FCM 具有最少的求得最优解的迭代次数. 在 SFT-SSVM-FCM 模型中, 在选择半正定核函数时的迭代次数和选择对称非半正定的核函数时的迭代次数是相同的. SSVM 模型和 SHSSVM 模型中的迭代总数几乎是相同的且最大差距不超过 0.3. 在 Iris 数据集上, 在选用“RBF”和“CS”核函数时, SFT-SSVM-FCM 的迭代总数分别为 1.1 和 1.3. SHSSVM 模型的迭代总数分别为 3.2 和 2.9. 而对于“RBF”核函数来说, SSVM 模

型的迭代总数为 2.9.

(4) 比起“RBF”核函数, 当选用“CS”核函数时, SFT-SSVM-FCM 模型分类精度更高, 训练时间更短, 但是迭代总数对于两种核函数来说是差不多的. 具体比较一下选用“CS”核函数和选用“RBF”核函数所对应的 SFT-SSVM-FCM 模型分类精度. 显然, 前者比后者要略高一些. 以 Banana 数据集为例: 选用“CS”核函数时模型分类精度比选用“RBF”核函数时的分类精度要高 1.15%. 对于训练时间的性能分析和分类精度的分析是类似的.

4.5 大规模数据集上的算法性能

最优子簇数量 c 由划分系数有效函数(23)来确定. 在后续实验中我们将采用吸收规则 3, 因为该结合规则在前述实验中表现性能良好. 同时, 接下来的实验中只是用高斯径向基核函数. 其中, 惩罚参数和核宽度参数值由 10-折交叉验证确定. 具体流程包含以下三个步骤.

(1) 确定参数选择范围.

(2) 测量算法性能. 将整个训练数据集分成 10 等份, 选择其中九份作为训练数据集构造出最终的决策函数, 将剩下的一份作为验证数据集用来测试由训练集得到的最优决策函数的泛化能力.

(3) 选择最优参数. 训练精度被定义为在 9 份训练集上的平均精度, 测试精度被定义为在验证数据集上的平均精度. 我们将验证数据集上最高分类精度对应的参数值作为最优参数值.

第一个实验在正态分布的二维数据集上进行, 训练集的大小从 200 变化到 2000. 正态分布的均值为 0.4, 方差为 0.6. 将 SFT-SSVM-FCM 模型和 SVM, SSVM 以及 HSSVM 模型作比较. 惩罚参数的值从 $\{0.01, 0.1, 0.3, 0.5, 0.8, 1\}$ 集合中选择, 核宽度参数的值从 $\{0.01, 0.1, 0.3, 0.5, 0.8, 1\}$ 中选择. 由于这些核宽度参数的值都可以使得分类精度高于 95%, 所以我们忽略分类精度随核宽度参数值的变化. 对于训练时间而言, 不同的算法在不同的核宽度参数值下的差异较大. 图 3 显示了不同算法的训练时间. 显然, SFT-SSVM-FCM 模型具有最短的训练时间, 并且这一优势随着训练数据集规模的增大而更加明显.

将区间 $[0, 1] \times [0, 1]$ 分割为 10000 个点, 生成棋盘格数据集, 并作为第二个实验的实验数据集. 核宽度参数 $\sigma \in \{0.1, 0.3, 0.8, 1, 3, 5\}$, 惩罚参数 C 从 $\{0.01, 0.5, 0.8, 1, 5, 10\}$ 中选择. 我们从这 10000 个点中选择样本点组成四个规模递增的训练数据集, 剩余样本作为测试集. 将 SFT-SSVM-FCM 模型和 SSVM、HSSVM、SHSSVM 和 SDWNSVM 模型作比较. 相关的算法性能列于表 7. 实验分别从分类精度, 训练时间和迭代总数来评估算法性能.

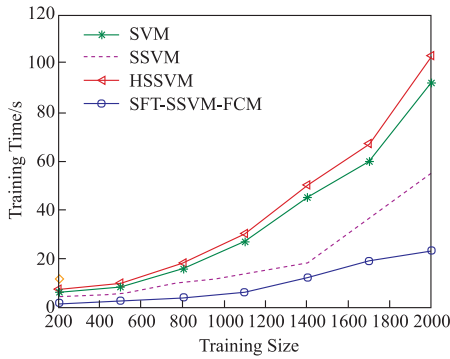


图3 不同算法的训练时间随训练集大小的变化

表 7 不同算法在不同数据集上的表现

Size	Algorithm	Accuracy	Time	Iteration
1000	SSVM	98.13%	4.56s	2.2
	HSSVM	96.42%	10.37s	/
	SHSSVM	97.82%	5.21s	2.1
	SDWNSVM	99.45%	4.10s	2.0
	SFT-SSVM-FCM	99.98%	2.16s	1.0
2000	SSVM	98.46%	11.97s	3.8
	HSSVM	99.18%	36.25s	/
	SHSSVM	98.99%	12.03s	4.1
	SDWNSVM	99.53%	9.67s	3.8
	SFT-SSVM-FCM	100%	4.69s	1.6
3000	SSVM	98.97%	25.21s	5
	HSSVM	97.91%	92.56s	/
	SHSSVM	99.07%	18.39s	5.3
	SDWNSVM	99.53%	23.35s	5.1
	SFT-SSVM-FCM	100%	10.88s	2.0
5000	SSVM	99.97%	86.54s	8
	HSSVM	99.18%	374.39s	/
	SHSSVM	99.07%	83.11s	8.6
	SDWNSVM	99.69%	78.27s	8.3
	SFT-SSVM-FCM	99.97%	22.12s	3.3

显然, SFT-SSVM-FCM 具有最高的分类精度, 最短的训练时间和最少的迭代次数. 尤其是在大样本数据集上, SFT-SSVM-FCM 减少训练时间的优势更加明显. 当训练集的大小为 1000 时, SFT-SSVM-FCM 的训练时间是 SSVM 的 47.37%, 是 HSSVM 的 20.81%, 是 SHSSVM 的 41.45%, 是 SDWNSVM 的 52.68%. 当训练集的大小为 5000 时, SFT-SSVM-FCM 的训练时间是 SSVM 的 25.56%, 是 HSSVM 的 5.9%, 是 SHSSVM 的 26.62%, 是 SDWNSVM 的 28.26%.

5 结论

本文详述了基于模糊 C-均值的相似性特征转换光滑支持向量机 (SFT-SSVM-FCM) 模型. 由于训练过程是在特征空间中进行的, 因此 SFT-SSVM-FCM 拓宽了核函数的选择范围. 在传统的 SVM 模型上使用平滑技术使得一些高效的无约束优化算法 (Newton-Armijo) 得以使用. 并且使用模糊 C-均值聚类技术将整个训练数据集划分成若干子簇, 在每个子簇上并行训练 SSVM 模型, 极大降低了训练时间. 实验结果也证明了 SFT-SSVM-FCM 模型比现有的基于 SVM 的模型和基于 SSVM 的模型在分类精度, 训练时间方面具有更好的表现. 测试阶段采用线性结合规则或者吸收规则来预测测试数据的标签值. SFT-SSVM-FCM 模型因其性能不随核宽度参数值的不同而发生明显变化, 所以具有较强的鲁棒性. 构造新的分片技术去划分训练集, 发掘新的结合规则以及寻找更高效的方法进行参数选择将是我们的下一步的研究目标.

参考文献

- [1] Vapnik V N. The Nature of Statistical Learning Theory [M]. New York: Springer-Verlag Press, 2000.
- [2] Cristianini N, Shawetaylor J. An Introduction to Support Vector Machines [M]. UK: Cambridge University Press, 2000.
- [3] ZHANG Li, ZHOU Wei-da, JIAO Li-cheng. Hidden space support vector machines [J]. IEEE Transactions on Neural Networks, 2004, 15(6): 1424 - 1434.
- [4] 王玲, 薄列峰, 刘芳, 焦李成. 稀疏隐空间支持向量机 [J]. 西安电子科技大学学报, 2006, 33(6): 896 - 901. WANG Ling, BO Lie-feng, LIU Fang, JIAO Li-cheng. Sparse hidden space support vector machine [J]. Journal of Xidian University, 2006, 33(6): 896 - 901. (in Chinese)
- [5] Dhanalakshmi P, Palanivel S, Ramal-Ingam, V. Classification of audio signals using SVM and RBFNN [J]. Expert Systems with Applications, 2009, 36(3): 6069 - 6075.
- [6] ZHANG Li, WANG Bang-jun, LI Fan-zhang, et al. Support vector novelty detection in hidden space [J]. Journal of Computational Information Systems, 2011, 15(7): 5581 - 5590.
- [7] 陈素根, 吴小俊. 改进的投影孪生支持向量机 [J]. 电子学报, 2017, 45(2): 408 - 416. CHEN Su-gen, WU Xiao-ju. Improved projection twin support vector machine [J]. Acta Electronica Sinica, 2017, 45(2): 408 - 416. (in Chinese)
- [8] 储茂祥, 王安娜, 巩荣芬. 一种改进的最小二乘孪生支持向量机分类算法 [J]. 电子学报, 2014, 42(5): 998 - 1003. CHU Mao-xiang, WANG An-na, GONG Rong-feng. Improvement on least squares twin support vector machine for

- pattern classification[J]. Acta Electronica Sinica, 2014, 42(5):998-1003. (in Chinese)
- [9] Lee Yuhjye, Mangasarian O L. SSVM: A smooth support vector machine for classification[J]. Computational Optimization and Applications, 2001, 22(1):5-21.
- [10] WU Qing, WANG Wen-qing. Piecewise smooth support vector machine for classification[J]. Mathematical Problems in Engineering, 2013, (3):301-312.
- [11] YUAN Yu-bo. Forecasting the movement direction of exchange rate with polynomial smooth support vector machine[J]. Mathematical and Computer Modelling, 2013, 57(3-4):932-944.
- [12] YUAN Yu-bo. Spline function smooth support vector machine for classification[J]. Journal of Industrial and Management Optimization, 2007, 23(5):529-542.
- [13] QIN Chuan-dong, LIU San-yang. Fuzzy smooth support vector machine with different smooth functions[J]. Journal of Systems Engineering and Electronics, 2012, 23(3):460-466.
- [14] 张晓丹, 邵帅, 刘钦圣. 基于样条函数的光滑支持向量机模型[J]. 北京科技大学学报, 2012, 34(6):718-725.
ZHANG Xiao-dan, SHAO Shuai, LIU Qin-sheng. Smooth support vector machine model based on spline functions [J]. Journal of University of Science and Technology Beijing, 2012, 34(6):718-725. (in Chinese)
- [15] SHEN Jin-dong, PENG Xiao-jun. A new smooth support vector machine with 1-norm penalty term[J]. International Journal of Online Engineering, 2013, 9(4):54-58.
- [16] LIANG Jin-jin, WU De. Smooth diagonal weighted newton support vector machine[J]. Mathematical Problems in Engineering, 2013, (2):1-10.
- [17] LUO H, LIU X, WANG S. Based on SARIMABP hybrid model and SSVM model of international crude oil price prediction research[J]. ANZIAM Journal, 2017, 58(E):E143-E161.
- [18] ZHANG X, CHEN J, et al. Modulation recognition of communication signals based on SCHKS-SSVM [J]. Journal of Systems Engineering and Electronics, 2017, 28(4):627-634.
- [19] XU J, Ramos S, Vázquez D, López A M. Hierarchical adaptive structural svm for domain adaptation[J]. International Journal of Computer Vision, 2016, 119(2):159-178.
- [20] LIANG Jin-jin, WU De. A hidden space smooth support vector machine with particle swarm optimization [J]. Journal of Xi'an Jiaotong University, 2013, 47(12):38-42.
- [21] LIANG Jin-jin, WU De. A smooth hidden space support vector machine[A]. Proceedings of the 9th International Conference on Natural Computation [C]. USA: IEEE Press, 2013. 1015-1020.
- [22] REN Shuang-qiao, YANG De-gui, LI Xiang, ZHUANG Zhao-wen. Piecewise support vector machines [J]. Chinese Journal of Computers, 2009, 32(1):77-85.
- [23] YE Qi-xiang, HAN Zhen-jun, JIAO Jian-bin, et al. Human detection in images via piecewise linear support vector machines [J]. IEEE Transactions on Image Processing, 2013, 22(2):778-789.
- [24] CHENG Jun, XIE Can, BIAN Wei, et al. Feature fusion for 3D hand gesture recognition by learning a shared hidden space[J]. Pattern Recognition Letters, 2012, 33(4):476-484.

作者简介



方佳艳 女, 1997 年出生, 安徽池州人. 电子科技大学信息与软件工程学院本科生, 主要研究方向为机器学习、人工智能.
E-mail: fyk80@163.com



吴德(通信作者) 男, 1979 年出生, 陕西西安人. 西安电子科技大学计算机学院博士, 高级工程师, 主要研究方向为机器学习、网络优化、数据挖掘.
E-mail: w00d1979@163.com