

文章编号: 1004-4353(2019)03-0215-06

基于改进粒子群优化算法和 CRNN 的多类 SVM 分类

俞颖^{1,2}, 黄风华^{1,2}, 阮奇³

(1. 阳光学院 空间数据挖掘与应用福建省高校工程研究中心;

2. 阳光学院 人工智能学院; 3. 阳光学院 教师发展中心; 福州 福建 350015)

摘要: 为了提高支持向量机(SVM)在 multi-class 分类中的分类效果,提出了一种基于改进粒子群优化(IMPSO)算法和协作式递归神经网络(CRNN)的多类 SVM 分类方法(IMPSO_CRNN_SVM 算法). 首先引入自适应惯性权重及自适应粒子变异,以此改进粒子群优化算法(PSO)在优化 SVM 参数过程中存在的容易陷入局部最优和早熟等问题;然后基于 multi-class SVM 设计一个 CRNN,并利用随机分配的训练集对该网络进行训练并构建最终决策函数,从而实现 multi-class 数据的“一次性”分类. 最后利用 3 种数据集和实际应用对 IMPSO_CRNN_SVM 算法进行验证,结果表明 IMPSO_CRNN_SVM 算法的分类精度优于未进行参数优化的传统 SVM 算法、基本 PSO 进行 SVM 参数优化的算法和未进行 PSO 参数优化的基于 CRNN 的多类支持向量机算法,因此 IMPSO_CRNN_SVM 算法具有一定的实用性.

关键词: 粒子群优化算法; 协作式递归神经网络; 支持向量机; 多类分类

中图分类号: TP301.6

文献标识码: A

Classification of multi-class support vector machines based on improved particle swarm optimization and CRNN

YU Ying^{1,2}, HUANG Fenghua^{1,2}, RUAN Qi³

(1. *Spatial Data Mining and Application Research Center of Fujian Province, Yango University*; 2. *Artificial Intelligence College, Yango University*; 3. *Teacher Development Centre, Yango University; Fuzhou 350015, China*)

Abstract: Aiming at the factors that affect the application of support vector machine (SVM) in multi-class classification, a multi-class SVM classification method (IMPSO_CRNN_SVM algorithm) based on improved particle swarm optimization algorithm (IMPSO) and cooperative recurrent neural network (CRNN) was proposed. Firstly, adaptive inertia weight and adaptive particle variation were introduced to improve the problem of local optimization and prematurity of particle swarm optimization algorithm (PSO) in the process of optimizing SVM parameters. Then, based on multi-class SVM technology, a CRNN was designed. The randomly assigned training set was used to train the network to construct the final decision function, so as to realize the “one-step” classification of multi-class data. Finally, the IMPSO_CRNN_SVM algorithm is verified by different data sets and practical applications. The results show that the classification accuracy of IMPSO_CRNN_SVM algorithm is better than that of SVM algorithm without parameter optimization or traditional PSO parameter optimization and multi-class SVM based on CRNN without parameter optimization, and it has certain practicability.

Keywords: particle swarm optimization; cooperative recurrent neural network; support vector machine; multi-class classification

0 引言

支持向量机(support vector machine,SVM)^[1]主要用于解决二类分类问题,而现实分类问题很多属于多类范畴,因此若将二类 SVM 应用在多类分类问题就需要对其进行参数优化和扩展.文献[2]采用蚁群算法对 SVM 进行了参数优化,该方法能在较短的时间内寻找到最优解,但计算量相对较大;文献[3]针对 SVM 优化对象设计了一种二进制编码基因串和相应的遗传算子,该方法虽然实现了 SVM 参数的组合优化,但需要解决个体基因的设计及编码问题;文献[4]利用蜂群的分工协作搜索最优蜜源来实现了 SVM 参数优化,该方法虽然可以实现局部和全局寻优,但时间开销较高;文献[5]提出了一种利用速度参数代替人工鱼步长的人工鱼群加速算法,该方法能够有效地克服参数优化后期难以逼近最优解的问题.文献[6-11]的作者利用改进的粒子群优化算法(PSO)对 SVM 的参数进行了优化,其研究结果均在一定程度上缓解了优化过程中可能出现的早熟和局部最优问题.

目前,将二类 SVM 扩展到多类分类场合的方法有两种:一是将原始的多类问题分解为多个二类问题,即通过构造多个二类分类器进行数据的“多次性”分类^[12];二是直接在目标函数上进行修改,将多个分类面的参数求解合并到一个最优化问题中,然后通过求解该最优化问题实现数据的“一次性”多类 SVM 分类^[13].第 1 种方法使用的算法主要有“一对一”和“一对多”,这两种算法虽然简单,但存在无法确定某些数据点的具体类别或者某些数据点被划为多个类别的现象.第 2 种方法的优点是可利用少量的支持向量获取较高的分类精度,但其在模型训练过程中需要求解一个复杂的优化问题.基于上述研究,针对二类 SVM 在多类分类场合的应用中存在的问题,本文通过引入自适应调节技术及协作式递归神经网络(CRNN),提出一种多类 SVM 分类方法,并通过实验验证该方法的有效性.

1 多类 SVM 的基本原理

1999 年,Bredensteiner 等提出了一种多类 SVM^[13].该多类 SVM 的基本原理是根据给出的已知 $K > 2$ 类分类样本点,构造一个在类之间进行有效区分的决策函数.该方法主要应用于 2 个以上类别的大数据集分类问题,其优点是可以通过少量的支持向量获得很好的分类效果,但需要求解一个较为复杂的优化问题.假设一个多类分类问题的 K 类数据集为:

$$(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N).$$

(1)

其中样本点 $\mathbf{x}_i \in \mathbf{R}^n (i=1,2,\cdots,N)$, n 为样本空间初始维度, N 为样本数量,每个样本的种类由 $y_i \in 1, 2, \cdots, K$ 表示,第 i 类样本数据集中包含样本点的数量为 m_i .多类 SVM 构造区分函数的目的是尽可能地区别出每一类与剩余类别,其基本形式为:

$$f_i(\mathbf{x}) = \omega^T \varphi(\mathbf{x}) - \gamma^i, \quad i=1,2,\cdots,K.$$

(2)

其中 $\varphi(\mathbf{x})$ 为特征向量由低维空间 \mathbf{R}^n 向高维空间 \mathbf{R}^M 的非线性映射,权重 $\omega \in \mathbf{R}^M$, 偏差 $\gamma \in \mathbf{R}$.决策函数的表达式为: $f(\mathbf{x}) = \max_{1 \leq i \leq K} \{f_i(\mathbf{x})\}$.以矩阵或向量形式细化公式(2),则最终的决策函数值可以通过计算以下 K 个区分函数的最大值来获取.

$$f_i(\mathbf{x}) = \frac{1}{K+1} \left\{ \sum_{j=1, j \neq i}^K \left[\sum_{\text{support vectors} \in \mathbf{A}^i} u_{ij}^T K(\mathbf{x}, \mathbf{A}_p^i) - \sum_{\text{support vectors} \in \mathbf{A}^i} u_{ji}^T K(\mathbf{x}, \mathbf{A}_p^i) \right] \right\} - \gamma^i,$$

(3)

其中 \mathbf{A}^i 为 $m_i \times n$ 维矩阵, \mathbf{A}_p^i 为对应矩阵的第 p 个样本点.支持向量 \mathbf{u} 的定义为

$$\mathbf{u}^T = [u^{12^T}, u^{13^T}, \cdots, u^{1K^T}, u^{21^T}, u^{23^T}, \cdots, u^{K(K-1)^T}].$$

对公式(3)中的 $K(\mathbf{x}, \mathbf{A}_p^i)$ 和 $K(\mathbf{x}, \mathbf{A}_p^i)$ 取径向基核函数,其基本形式为 $K(\mathbf{x}, \mathbf{y})$, 具体定义如下:

$$K(\mathbf{x}, \mathbf{y}) = \exp(-g \|\mathbf{x} - \mathbf{y}\|^2).$$

2 IMPSO 优化多类 SVM 参数

为了有效地克服传统 PSO 算法在 SVM 参数优化过程中容易出现的早熟收敛及陷入局部最优问题,本文采用自适应惯性权重结合自适应粒子变异来动态调节粒子的进化过程,以此实现对 PSO 算法的改进(该算法记为 IMPSO).

假设在一个 D 维空间中,粒子 i 的位置以 $\mathbf{X}_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD})$ 表示,速度以 $\mathbf{V}_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{iD})$ 表示.在每一次迭代中,粒子 i 通过跟踪个体极值 \mathbf{pbest}_i 和群体极值 \mathbf{gbest} 来更新自身的速度和位置. \mathbf{pbest}_i 表示粒子自身找到的最优解,可以表示为 $\mathbf{pbest}_i = (p_{i1}, p_{i2}, p_{i3}, \dots, p_{iD})$; \mathbf{gbest} 表示整个群体所找到的最优解,可以表示为 $\mathbf{gbest} = (g_1, g_2, g_3, \dots, g_D)$.

在第 d ($1 \leq d \leq D$) 维空间中,粒子的速度和位置的更新公式如下:

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 (pbest_{id} - x_{id}^k) + c_2 r_2 (gbest_d - x_{id}^k), \quad (4)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1}. \quad (5)$$

其中 ω 为惯性权重, v_{id}^k 为第 k 次迭代粒子 i 速度矢量的第 d 维分量, x_{id}^k 为第 k 次迭代粒子 i 位置矢量的第 d 维分量, c_1 和 c_2 为学习因子, r_1 和 r_2 为在 $[0, 1]$ 区间分布的随机参数.

2.1 自适应惯性权重

自适应惯性权重的优化公式^[14]为:

$$\omega = \begin{cases} \omega_{\min} - \frac{(\omega_{\max} - \omega_{\min}) \times (f - f_{\min})}{f_{\text{avg}} - f_{\min}}, & f \leq f_{\text{avg}}; \\ \omega_{\max}, & f > f_{\text{avg}}. \end{cases} \quad (6)$$

其中 ω_{\min} 和 ω_{\max} 为惯性权重的最小值和最大值, f 为粒子当前的目标适应度值, f_{avg} 和 f_{\min} 分别表示当前所有粒子的平均适应度值和最小适应度值.自适应惯性权重 ω 能够随目标适应度值的变化而进行动态调整,在优化过程中可避免出现局部极小值,进而提高算法的搜索能力^[15].

2.2 自适应粒子变异

为了实现动态调节每代粒子变异概率,本文在迭代过程中采用如下种群粒子概率变异方式:

- 1) rand: 产生随机变异概率
- 2) 如果 $\text{rand} > 0.5$
- 3) $k = \text{ceil}(2 * \text{rand})$
- 4) if $k == 1$
- 5) $x_{ik} = (20 - 1) * \text{rand} + 1$
- 6) endif
- 7) if $k == 2$
- 8) $x_{ik} = (x_{g\max} - x_{g\min}) * \text{rand} + x_{g\min}$
- 9) endif

其中 x_{ik} 为粒子 i 第 k 维的位置, g 的区间为 $[x_{g\min}, x_{g\max}]$.

3 基于 CRNN 的多类 SVM 分类

CRNN 由若干个递归神经网络(RNN)组成,其对解决约束性优化问题具有明显的优势^[16-17].为了更高效地对“一次性”多类 SVM 分类问题进行求解,本文将原始多类 SVM 分类问题进行分解,并设计一个 CRNN 用于多类 SVM 学习,以此实现分类.本文将原始多类 SVM 分类问题分解成 2 个子问题,即求解支持向量 \mathbf{u} 的二次规划问题和求解偏差 γ 的线性规划问题:

$$\min \varphi(\boldsymbol{u}) = \frac{1}{2} \boldsymbol{u}^T \boldsymbol{Q} \boldsymbol{u}^T - \boldsymbol{e}^T \boldsymbol{u}, \tag{7}$$

$$\text{s. t. } \bar{\boldsymbol{E}}^T \boldsymbol{u} = 0, \quad 0 \leq \boldsymbol{u} \leq \boldsymbol{h};$$

$$\min \boldsymbol{c}_u^T \boldsymbol{z}, \tag{8}$$
$$\text{s. t. } \boldsymbol{B} \boldsymbol{z} = \boldsymbol{b}, \quad \boldsymbol{z} \geq 0.$$

其中: $\boldsymbol{Q} = \frac{1}{K+1} \bar{\boldsymbol{A}} \bar{\boldsymbol{A}}^T$; $\boldsymbol{e} = (1, \cdots, 1) \in \mathbf{R}^{(K-1) \sum_{i=1}^K m_i}$; $\boldsymbol{h} = \frac{1-\lambda}{\lambda} \boldsymbol{e}$, $0 < \lambda < 1$; $\boldsymbol{c}_u = \begin{pmatrix} \boldsymbol{Q} \boldsymbol{u} - \boldsymbol{e} \\ \boldsymbol{o}_1 \end{pmatrix}$; $\boldsymbol{B} = \begin{pmatrix} -\bar{\boldsymbol{E}} & -\boldsymbol{I} \\ \boldsymbol{O} & -\boldsymbol{I} \end{pmatrix}^T$, $\boldsymbol{b} = \begin{pmatrix} \boldsymbol{o}_2 \\ -\boldsymbol{e} \end{pmatrix}$, \boldsymbol{I} 为单位矩阵, \boldsymbol{o}_1 、 \boldsymbol{o}_2 和 \boldsymbol{O} 分别是相应行数的零向量. 由此, 原始问题的求解被分解为对 2 个约束性优化问题的求解. 上述中 $\bar{\boldsymbol{E}}$ 和 $\bar{\boldsymbol{A}}$ 的定义参见文献[13].

本文设计的 CRNN 由 2 个 RNN 自适应构造而成, 其中一个 RNN 用于求解二次规划问题(7), 记为 RNN(9); 另一个 RNN 用于求解线性规划问题(8), 记为 RNN(10).

$$\frac{d}{dt} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{\beta} \end{pmatrix} = \eta \begin{pmatrix} \boldsymbol{P}_x(\boldsymbol{u} - (\boldsymbol{Q} \boldsymbol{u} + \bar{\boldsymbol{E}} \boldsymbol{\beta} - \boldsymbol{e})) - \boldsymbol{u} \\ -\bar{\boldsymbol{E}}^T \boldsymbol{u} \end{pmatrix}, \tag{9}$$

$$\frac{d}{dt} \begin{pmatrix} \boldsymbol{z} \\ \boldsymbol{v} \end{pmatrix} = -\eta \begin{pmatrix} (\boldsymbol{c}_u^T \boldsymbol{z} - \boldsymbol{b}^T \boldsymbol{v}) \boldsymbol{c}_u + \boldsymbol{B}^T (\boldsymbol{B} \boldsymbol{z} - \boldsymbol{b}) - (-\boldsymbol{z})^+ \\ -(\boldsymbol{c}_u^T \boldsymbol{z} - \boldsymbol{b}^T \boldsymbol{v}) \boldsymbol{b} + \boldsymbol{B} (\boldsymbol{B}^T \boldsymbol{v} - \boldsymbol{c}_u)^+ \end{pmatrix}. \tag{10}$$

其中 η 是一个正的参数, $\boldsymbol{\beta} \in \mathbf{R}^K$, $\boldsymbol{z} \in \mathbf{R}^{(K-1) \sum_{i=1}^K m_i}$, $\boldsymbol{v} = [\gamma, y]^T$. 将 $\boldsymbol{P}_x(u_i)$ 定义为:

$$\boldsymbol{P}_x(u_i) = \begin{cases} 0, & u_i < 0; \\ u_i, & 0 \leq u_i \leq h_i; \\ h_i, & u_i > h_i. \end{cases} \tag{11}$$

其中 $i=1, \cdots, (K-1) \sum_{i=1}^K m_i$. 利用 RNN(9) 和 RNN(10) 求解上述 CRNN 即可获取支持向量 \boldsymbol{u} 及偏差 γ 的全局最优解, 进而可以通过计算决策函数(3) 的最大值来一次性获取多类 SVM 分类的结果.

4 算法的流程

本文提出的多类 SVM 分类的算法流程为:

- 1) 准备原始分类数据集 Data, 将其随机划分为训练集 Data1 和测试集 Data2, 并对训练集和测试集行归一化处理;
- 2) 设定 IMPSO 算法的原始参数, 包括迭代次数、种群规模、活动区间以及 c_1 、 c_2 、 r_1 和 r_2 等;
- 3) 随机产生粒子的位置和速度, 设置惯性权重的初始值, 计算初始适应度值;
- 4) 自适应更新粒子的惯性权重和更新粒子的速度和位置, 然后对粒子进行自适应变异, 并计算其适应度值;
- 5) 更新个体极值 \boldsymbol{pbest}_i 和群体极值 \boldsymbol{gbest} ;
- 6) 判断是否超过最大迭代次数, 如果是则获取最优多类 SVM 参数, 转步骤 7); 否则转步骤 4), 继续迭代;
- 7) 利用所设计的 CRNN 对问题求解: 利用 RNN(9) 求解多类 SVM 的支持向量 \boldsymbol{u} , 利用 RNN(10) 求解多类 SVM 的偏差 γ ;
- 8) 构造决策函数(3), 通过计算函数(3) 的最大值对待分类数据进行“一次性”分类;
- 9) 利用分类结果计算分类精度和分析分类性能.

5 算法的实验验证

算法在 Matlab R2014a 环境下编程实现. 为了验证本文所提出算法的性能,将本文提出分类算法的性能(记为 IMPSO_CRNN_SVM)分别与其他 3 种算法(未进行参数优化的原始 SVM 算法(记为 SVM)、基本 PSO 进行 SVM 参数优化的算法(记为 PSO_SVM)、未进行 PSO 参数优化基于 CRNN 的多类支持向量机(记为 CRNN_SVM))进行对比.

实验中使用 UCI 数据库中的 Iris、Wine 及 abalone 3 个数据集对各算法的分类性能进行验证. 表 1 为 3 种数据集的基本信息,表 2 为 4 种算法在相同数据集上测试所达到的分类精度(算法在同一数据集上连续运行 5 次所得出的精度平均值). 表 3 为 CRNN_SVM 算法和 IMPSO_CRNN_SVM 算法在 Iris 及 Wine 2 种数据集上的运行时间效率对比. 这 2 种算法的迭代次数为 300,种群数目为 30,学习因子 $c_1=1.6$, $c_2=1.5$.

表 1 实验数据集的信息表

数据集	粒子数目	特征数目	训练集数目	测试集数目
Iris	30	4	48	102
Wine	30	12	89	89
abalone	30	9	900	3 277

表 2 不同算法的分类精度

数据集	IMPSO_CRNN_SVM		CRNN_SVM		SVM		PSO_SVM	
	训练集精度	测试集精度	训练集精度	测试集精度	训练集精度	测试集精度	训练集精度	测试集精度
Iris	1.000 0	0.979 2	1.000 0	0.927 0	1.000 0	0.927 0	1.000 0	0.947 9
Wine	1.000 0	0.955 1	1.000 0	0.943 8	1.000 0	0.932 6	1.000 0	0.955 1
abalone	0.888 9	0.884 0	0.822 2	0.884 0	0.775 6	0.756 7	0.856 7	0.884 0

表 3 不同算法的运行时间

算法	数据集	SVM 参数优化耗时	训练耗时	测试耗时
CRNN_SVM	Iris	0.000 0	3.251 4	0.067 2
	Wine	0.000 0	1.149 6	0.098 6
IMPSO_CRNN_SVM	Iris	6.059 6	2.401 6	0.029 5
	Wine	15.639 1	1.075 2	0.095 7

由表 2 可以看出,PSO_SVM 算法和 IMPSO_CRNN_SVM 算法的分类精度总体上高于 SVM 算法和 CRNN_SVM 算法,其中本文提出算法的分类精度相对最高. 由表 3 可以看出,本文提出的算法在优化 SVM 参数时需要的时间相对略长,但分类器的训练时间低于 CRNN_SVM 算法. CRNN_SVM 算法的训练时间相对较长的原因是其参数(经验值)的设置需要经过多次试验才能获取.

6 算法应用

为了进一步验证本文算法的有效性和实用性,将 IMPSO_CRNN_SVM 算法应用于遥感影像分类. 验证选取的是 1 幅多光谱遥感影像图,图中包涵水域、林地、草地和裸地 4 个类别的数据(图 1). 该影像图共有 6 个波段,数据集的基准是 wgs-84,影像的空间分辨率是 30 m.

首先对遥感影像图进行目视解译,并在实验区

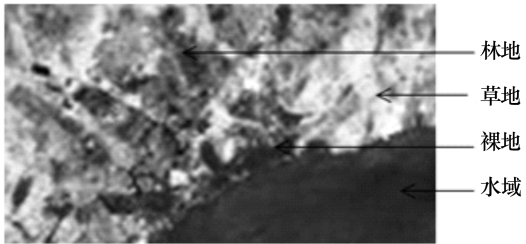


图 1 原始遥感影像图

中提取 4 个类别的样本点数据, 每类数据量为 300 个, 共计 1200 个. 然后将每类数据的样本点随机划分为训练集(50 个样本点)和测试集(250 个样本点), 并对其进行归一化处理. 最后利用 IMPSO_CRNN_SVM 算法进行分类处理. 通过 IMPSO 优化之后的 SVM 参数取值为: $bestc = 74.5905$, $bestg = 0.0100$. 由 CRNN 构造决策数据得到的分类实验结果为: 训练集正确率为 100%, 测试集正确率为 97.7%. 由此可以看出, 本文提出的 IMPSO_CRNN_SVM 算法具有良好的分类效果.

利用本文提出的 IMPSO_CRNN_SVM 算法、CRNN_SVM 算法、PSO_SVM 算法及 SVM 算法对实验区遥感影像进行分类识别, 结果如图 2—图 5 所示.

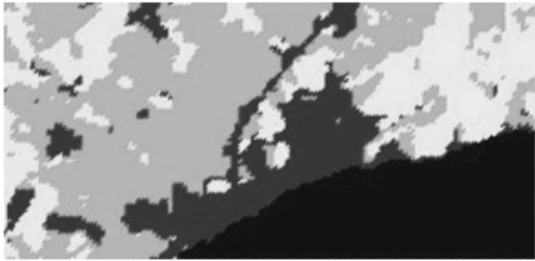


图 2 基于 IMPSO_CRNN_SVM 算法的影像分类

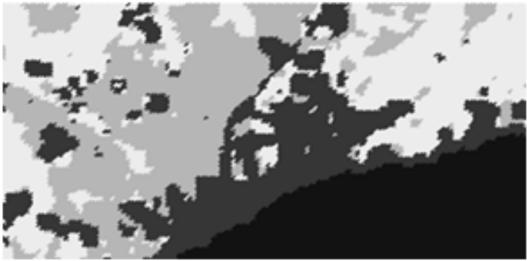


图 3 基于 CRNN_SVM 算法的影像分类

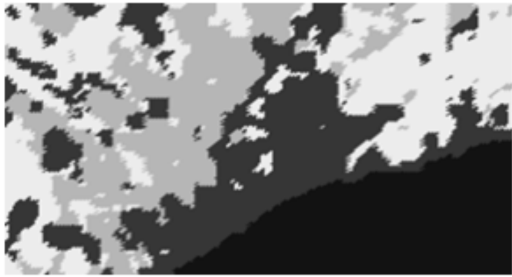


图 4 基于 PSO_SVM 算法的影像分类

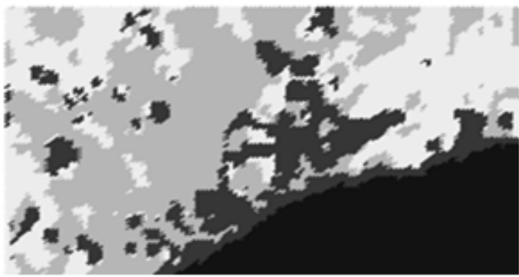


图 5 基于 SVM 算法的影像分类

由图 2—图 5 可以看出, IMPSO_CRNN_SVM 算法在边界处理方面整体优于其他 3 种算法. 其中: SVM 算法对裸地的识别精度低于 CRNN_SVM 和 PSO_SVM 算法, 但 SVM 算法对林地的识别精度优于 CRNN_SVM 和 PSO_SVM 算法; IMPSO_CRNN_SVM 算法对林地的识别精度较高, 但对裸地的识别精度相对较低.

7 结论

研究表明, 本文提出的 IMPSO_CRNN_SVM 算法的分类精度优于未进行参数优化的 SVM 算法、基于传统 PSO 的 SVM 算法及未进行参数优化的基于 CRNN 的多类 SVM 分类方法, 且可有效地避免粒子群优化算法在优化多类 SVM 参数过程中容易陷入局部最优以及“多次性”SVM 分类可能存在的不可分区域的问题, 因此 IMPSO_CRNN_SVM 算法具有很好的实用价值. 在今后的研究中, 我们将利用分布式并行处理技术进一步降低算法的计算代价, 提高算法的训练效率.

参考文献:

[1] CORINNA Cortes, VLADIMIR Vapnik. Supprot vector networks[J]. Machine Learning, 1995,20(3):273-297.
[2] 庄严,白振林,许云峰. 基于蚁群算法的支持向量机参数选择方法研究[J]. 计算机仿真,2011,28(5):216-219.
[3] 刘靖洁,陈桂明,刘小方,等. 基于遗传算法的 SVM 参数组合优化[J]. 计算机应用与软件,2012,29(4):94-96.
[4] 李楠,朱秀芳,潘耀忠,等. 人工蜂群算法优化的 SVM 遥感影像分类[J]. 遥感学报,2018,22(4):559-569.
[5] 邱云飞,李智义. 改进人工鱼群算法在 SVM 参数优化中的应用[J]. 计算机工程与科学,2018,40(11):2076-2078.

而导致将不同的人脸图像聚为一类的问题,并且还可以提高获取人脸数据集的时间效率,因此本文方法可被广泛应用在人脸识别、人脸数据库的建立以及人脸搜索等系统中.由于人脸聚类算法中的特征提取阶段较为耗时,使得本文方法的时间效率低于基于人脸跟踪的人脸数据集生成方法.另外,由于本文方法未能对完全侧脸的人脸图像进行正确聚类,导致生成的人脸数据集的个数可能多于实际个数.在今后的研究中,我们将对上述问题进行研究,以完善本文方法.

参考文献:

[1] 李蕊岗,张明. 基于改进 Camshift 和 Kalman 滤波的目标跟踪算法[J]. 微型机与应用,2017,36(12):39-41.

[2] HENRIQUES J F, CASEIRO R, MARTINS P, et al. High-speed tracking with Kernelized Correlation Filters[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015,37(3):583-596.

[3] NAM Hyeonseob, HAN Bohyung. Learning multi-domain convolutional neural networks for visual tracking[C]// Computer Vision and Pattern Recognition. Las Vegas, USA: CVPR, 2016:4293-4302.

[4] SAHOO S K, MAKUR A. Dictionary training for sparse representation as generalization of K-means clustering [J]. IEEE Signal Processing Letters, 2013,20(6):587-590.

[5] 陈恩. 人脸深度特征获取及聚类技术研究[D]. 杭州:浙江工业大学,2017.

[6] 王玉晗,罗邓三郎. 聚类算法综述[J]. 科技资讯,2018,16(24):10-11.

[7] BIEMANN Chris. Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems[C]//The First Workshop on Graph Based Methods for Natural Language Processing. Stroudsburg: Association for Computational Linguistics, 2006:73-80.

[8] 张乘龙,夏筱筠,柏松,等. 基于 KCF 跟踪算法的目标轨迹记录系统[J]. 计算机系统应用,2017,26(5):113-118.

[9] HE Kaiming, ZHANG Xiangyu, REN Shaoqing, et al. Deep residual learning for image recognition[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). New York: IEEE, 2016:770-778.

—————
(上接第 220 页)

[6] SUBAI A. Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders[J]. Computers in Biology and Medicine, 2013,43(5):576-586.

[7] 王振武,孙佳骏,尹成峰. 改进粒子群算法优化的支持向量机及其应用[J]. 哈尔滨工程大学学报,2016,37(12):1728-1733.

[8] 张进,丁胜,李波. 改进的基于粒子群优化的支持向量机特征选择和参数联合优化算法[J]. 计算机应用,2016,36(5):1330-1335.

[9] 于梦馨,刘波,汤恩生. 改进粒子群算法优化 SVM 参数的遥感图像分类[J]. 航天返回与遥感,2018,39(2):133-140.

[10] 张国梁,贾松敏,张祥银,等. 采用自适应变异粒子群优化 SVM 的行为识别[J]. 光学精密工程,2017,25(6):1669-1678.

[11] 赵志刚,黄树运,王伟倩. 基于随机惯性权重的简化粒子群优化算法[J]. 计算机应用研究,2014,31(2):361-364.

[12] HSU C W, LIN C J. A comparison of methods for multi-class support vector machines[J]. IEEE Transaction on Neural Network, 2002,13(2):415-425.

[13] BREDENSTEINER E J, BENNETT K P. Multicategory classification by support vector machines[J]. Computational Optimization and Applications, 1999,12(1):53-79.

[14] 于梦馨,刘波,汤恩生. 改进粒子群算法优化 SVM 参数的遥感图像分类[J]. 航天返回与遥感,2018,39(2):133-140.

[15] 张俊红,刘昱,马文朋,等. 基于 GAPSO-SVM 的航空发动机典型故障诊断[J]. 天津大学学报,2012,45(12):1057-1061.

[16] HU Z X, HU Y M, LIU J, et al. A CRNN module for hand pose estimation[J]. Neurocomputing, 2019,333:157-168.

[17] LI W, CAO L, ZHAO D, et al. CRNN: Integrating classification rules into neural network[C]//2013 International Joint Conference on Neural Networks. Dallas: IEEE, 2013.