

基于元学习和二进制粒子群的网络故障特征选择算法

卫娟, 王崇科

(河南机电高等专科学校, 河南 新乡 453002)

摘要: 为了降低 Wrapper 模式网络故障特征选择方法分类算法的计算量, 文章提出了一种基于元学习和二进制粒子群 (ML-BPSO) 的特征选择方法; 算法在封装的分类训练中采用元学习方法估算分类精度, 并利用 BPSO 在特征空间中进行全局搜索选出最优特征集; 在 DARPA 数据集上的实验可以看出本文方法选取结果与 BPSO-SVM 相当但是计算量大大降低; 实验结果表明文章提出的方法能够显著的降低网络故障特征选择计算量, 同时保证了较高的诊断精度和较好的降维效果。

关键词: 元学习; 二进制粒子群; 支持向量机; 特征选择

A Meta-learning and Binary Particle Swarm Optimization Based Network Fault Feature Selection Algorithm

Wei Juan, Wang Chongke

(Henan Mechanical and Electrical Engineering College, Xinxiang 453002, China)

Abstract: The wrapper network fault feature selection algorithms get large calculation cost, a Meta-learning and binary particle swarm optimization (ML-BPSO) based feature selection algorithm was proposed to solve this problem in this paper. The Meta-learning method was introduced for estimating the classification accuracy wrapped in selected method. On this basis, the BPSO is used for searching the whole feature space to find the best feature subset. The experiment on DARPA datasets shows the proposed method result approximate to BPSO-SVM and the calculation cost reduced expressly. The result shows ML-BPSO reduce the calculation cost while gets good performance on classification accuracy and dimensional decrease.

Keywords: meta-learning; BPSO; support vector machine; feature selection

0 引言

当前, 网络故障诊断方法往往将收集的故障数据直接进行训练和识别。但是这些原始数据中往往含有冗余甚至噪声特征, 这些特征不仅会增加训练的复杂度, 还可能降低诊断精度, 在诊断过程中, 还会影响诊断速度。可见从采集的初始故障集中选出最能代表故障特性的子集对诊断具有十分重要的意义。

近年来, 特征选择问题得到了广泛的研究, 根据是否依赖机器学习算法, 特征选择算法可以分为三大类^[1-3]: Filter 型算法^[4-5]、Wrapper 型^[6]以及 Embedded 型算法^[7]。其中, Wrapper 型算法能够获得最佳的选择效果, 但是它的计算代价要高于其他两种方法。本文选择 Wrapper 型算法, 采用 SVM 作为它的机器学习算法。前面已经提到 Wrapper 最大的缺点是计算代价高, 在特征选择过程中需要进行大量的训练, 阻碍了这些方法的实用性。因此, 如何降低 Wrapper 型特征选择的计算代价成为亟待解决的问题。

本文分析了基于 SVM 的典型 Wrapper 算法, 通过元学习方法估算分类器的分类精度避免训练, 降低运算量。并结合二

进制粒子群 (BPSO) 算法寻找最优的特征组合, 提出一种新颖的基于元学习和 BPSO 的网络故障特征选择算法。

1 相关研究工作分析

网络故障特征选择实际上是一个组合优化问题, 可以采用一些启发式的搜索算法求解, 关键是如何设置优化目标 F 。文献 [6] 提出的 BPSO-SVM 特征算法是一种非常典型的 Wrapper 算法。在该算法中, 优化的目标函数综合考虑分类准确性 (accuracy) 和特征维数 (feature_dim):

$$F = (1/2)accuracy + (1/2)(1/feature_dim) \quad (1)$$

其中: SVM 的 accuracy 是通过交叉验证的方式得到的。计算 accuracy 过程中, 若采用文献 [6] 所提的采用 5 折交叉验证, 则对于每个选取的参数均需要进行 5 次 SVM 训练才能确定最终的 accuracy。观察 (1) 式我们可以发现在整个目标函数中, 所有的计算量几乎全部在计算 accuracy 上, 它的获得需要对样本进行训练和测试。如果能够直接估算出 accuracy, 那么将可以极大的降低计算量。

为了解决这一问题, 我们引入元学习的思想。元学习方法通过对学习结果的再学习以提高机器学习效果, 是一种先进的机器学习理论, 已经在模式识别、预测等领域获得了成功^[8-9]。一些学者对元学习理论进行了拓展: 通过分析已有数据集的组成和统计特性等构造元特征, 建立元特征与分类结果的映射关系构造元学习模型, 在目标样本集中估算学习结果, 大幅降低运算量^[10-11]。国外学者已成功将其应用于学习机器的参数选择、样本选择等领域, 然而国内对于这种思路关注非

收稿日期: 2014-08-03; 修回日期: 2014-09-18。

基金项目: 河南省高等学校青年骨干教师资助计划 (2011GGJS-198); 河南省教育厅科学技术研究重点项目 (13A520221)。

作者简介: 卫娟 (1980-), 女, 河南获嘉人, 硕士, 讲师, 主要从事计算机网络、软件工程、算法设计等方向的研究。

常少。本文采用后一种思路对诊断精度进行估算，在此基础上实现网络故障特征选择。

2 算法描述

通过第二部分的分析，本文提出的基于元学习和二进制粒子群的网络故障特征选择算法流程如图 1 所示。首先选择一些已知的公用数据集，这些数据集构造元特征，并对它们分别进行分类训练，训练方法采用 SVM，在训练过程中对 SVM 进行设计（参数进行寻优，对模型进行优化），得到结果比较好的模型，统计它们的分类精度。将构造的元特征和分类精度组合配对，构建元数据集。对构造的元数据集进行回归学习，建立元特征与分类精度之间的元模型。在此基础上，对故障数据集进行特征选择，搜索算法采用二进制粒子群，对特征进行相应的编码，根据编码情况选择特征并构造相应的元特征，将所得的元特征带入元模型中估算每次的诊断精度，进而进行进一步的更新、寻优直至找出最佳故障特征子集。

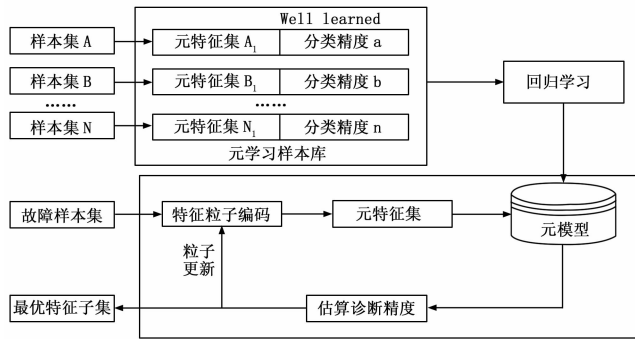


图 1 ML-BPSO 特征选择算法框图

算法的整个过程主要包括两个部分：元学习模型的构建以及最优故障特征子集的搜索选择。下面就这两部分分别进行详细介绍。

2.1 元学习模型构建

元学习模型的构建是算法的核心，它主要包括两个步骤：元学习样本库的建立和元模型构建。元特征集是从数据集中提取出的，用以描述整个数据集特征的一些参数，例如样本个数、样本的维数、类别数量、信息熵等等，它们直接决定了元学习模型建立的好坏。本文采用文献 [8] 所使用的元特征集，从大类上可以分为：简单元特征（如个数、维数）、统计元特征（如样本集的峰值、协方差等）、基于信息论的元特征（如信息熵）、基于模型的元特征等共 17 维。

分类训练采用 SVM 方法，通过网格搜索法确定最优参数，5 折交叉验证获得最终的分类精度，结合前面抽取的元特征构建出元学习样本集。在此基础上，利用 ϵ -SVR 回归算法，得到元特征集与分类精度之间的回归元学习模型。在随后的特征搜索过程中，只需要将目标样本集的元特征集抽取出来，输入到元学习模型中就可估算出此特征集的分类精度。

2.2 BPSO 算法

粒子群算法是 Kennedy 和 Eberhart 模仿鸟类群体行为的智能优化算法，可解决连续函数的优化问题^[12]。在算法中，群体中的每个粒子都是一个潜在的解，通过学习历史中自身的最优位置 p_b 和群体最优位置 p_g 来更新位置和速度，并根据粒子的位置计算适应度函数来判断解的优劣，不断迭代找到最

优解。

若粒子群的搜索空间为 D 维，粒子的位置和速度分别为 $P_i = (p_{i1}, p_{i2}, \dots, p_{iM})$ 和 $V_i = (v_{i1}, v_{i2}, \dots, v_{iM})$ ，则其迭代公式为：

$$\begin{cases} v_i^{t+1} = v_i + c_1 r_1 (p_b - p_i) + c_2 r_2 (p_g - p_i) \\ p_i^{t+1} = p_i + v_i^{t+1} \end{cases} \quad (2)$$

其中： t 为迭代次数， c_1 和 c_2 为学习因子， r_1 和 r_2 为 $[0,1]$ 之间的随机数。

为解决粒子群在离散问题中的应用，Kennedy 和 Eberhart 又在标准粒子群的基础之上提出了二进制粒子群优化算法^[10]。其原理和速度的更新方式不变，只是将粒子位置的每一维分量限制为 0 或 1，并根据速度的 sigmoid 函数变换来控制粒子的位置更新：

$$\begin{aligned} & \text{if } \text{rand} < s(v_{im}) \\ & \text{then } p_{im}^{t+1} = 1; \text{else } p_{im}^{t+1} = 0 \end{aligned} \quad (3)$$

本文提出的 ML-BPSO 算法采用 wrapper 特征选择模型，利用 BPSO 的自动寻优能力在故障特征空间中进行全局搜索，得到不同特征组合。根据 SVM 分类结果判断这些特征组合的分类性能，并不断更新选取的特征集，直至搜索到取得最佳分类结果的特征组合。与神经网络、遗传算法等优化算法相比，PSO 具有所需确定的参数较少，收敛速度更快等优点；与顺序选择算法等常用特征选择方法相比，PSO 不易陷入局部极值，能得到全局最优解。

2.3 ML-BPSO 故障特征选择算法

如果将整个故障特征空间看作解空间，不同特征组合看作解空间中不同位置处的粒子，粒子中各位置分量取值对应特征组合中各特征分量的状态。在 BPSO 寻优中，首先需要根据问题对候选解进行编码：把每一个特征定义为粒子的二进制变量，粒子空间维数 D 由原始特征集维数决定。如果第 i 位为 1，那么第 i 个特征就被选中，否则这个特征就被丢弃。

BPSO 算法中，某个粒子的位置矢量决定了某种特征子集的组合方式，而整个群体最优位置 P_g 确定了系统的最优特征组合，适应度函数 Fit 指导粒子群搜索方向。特性选择的目的是使用尽可能少的特征得到相同或更优的分类性能，因此，适应度函数需要综合考虑分类准确性 (accuracy) 和特征维数 (feature_dim)。式 (1) 给出的适应度函数将这两部分视作同等重要，但是当 accuracy 较小时 (如 0.7)，若 feature_dim 为 1，则 $F = 0.35 + 0.5 = 0.85$ ，此时的 F 高于当 accuracy 达到 1 而 feature_dim 为 2 ($F = 0.5 + 0.25 = 0.75$)，显然，我们不能认为第一种情况好于第二种情况。其它文献给出的目标函数同样没有很好的解决这两部分的权值关系。若原始样本集训练获得的分类精度 (accuracy_1)，在以下假设的前提下：

假设：经过特征选择后获得的最终 accuracy 同 accuracy_1 相差不大。

对大量数据集的实验可以看出经过特征选择得到的最终 accuracy 一般略高于或者略低于 accuracy_1，这一假设是成立的。另外，考虑在目标函数中，我们更注重最终的 accuracy，且认为当 feature_dim 对目标函数的影响较 accuracy 低一个数量级时由 accuracy 主导目标函数值。我们给出的最终的目标函数为：

$$Fit = accuracy + 0.1accuracy_1(1/feature_dim) \quad (4)$$

结合 2.1、2.2，给出 BPSO 进行特征选择的步骤：

输入: 初始样本集 $S = SV = \{(x_i, y_i) \mid x_i \in \mathbf{R}^d, y_i \in \{1, 2, \dots, c\}, i = 1, \dots, l_{sv}\}$ (c 为类别数, l_{sv} 为支持向量个数), 种群规模 n_p , 最大迭代次数 $iter$, 适应度函数 Fit , 阈值 Th 。

输出: 特征提取后的样本集 $S' = \{(x_i, y_i) \mid x_i \in \mathbf{R}^d, y_i \in \{1, 2, \dots, c\}, i = 1, \dots, l_w\}$ 。

步骤 1: 初始化粒子位置和速度, 设定粒子规模 n_p , 最小和最大飞行速度 v_{min} 和 v_{max} , 最大迭代次数 $iter$, 适应度函数阈值 Th , 计算每个粒子适应度函数 Fit , 初始化 p_b 、 p_g 以及迭代次数 n_p 。

步骤 2: for $i = 1$ to n_p ;

计算元特征, 根据回归模型估算 accuracy, 并由此计算出 Fit

$$\text{if } Fit(x_i) > Fit(p_i) \quad p_b = x_i$$

$$\text{if } Fit(x_i) > Fit(p_g) \quad p_g = x_i$$

步骤 3: for $i=1$ to n_p ;

for $i=1$ to D

根据式 (2) 和 (3) 更新每个粒子的速度 v_{id} 和位置 x_{id}

$n = n + 1$

步骤 4: for $i = 1$ to n_p

$$\text{if } (n < iter) \ \& \ (Fit(x_i) < Th)$$

返回 2

else 停止迭代, 输出群体最佳位置 p_g 和 $Fit(p_g)$ 。

步骤 5: 选择当前种群中最优个体 p_g 中为 1 的基因位对应 S 中的样本特征, 构成约简故障特征集 $S' = \{(x_i, y_i) \mid x_i \in \mathbf{R}^d, y_i \in \{1, 2, \dots, c\}, i = 1, \dots, l_{sv}\}$ 输出。

2.4 算法验证

为了验证所提出的特征选择算法的有效性, 采用 UCI 数据集对其进行测试^[13], 表 1 给出了这些数据集的概要描述。由这些数据集的多元性可知, 它们在一定程度上能够验证特征选择算法在不同条件下的性能。另外, 由于样本特征属性值之间的不同度量, 使得样本向量在计算距离时取值范围偏大的属性占据了主导地位, 大大弱化了取值范围偏小的特征的贡献, 无法正确反映样本间的真实差异。设特征集合 $x = \{x_1, x_2, \dots, x_d\}$ 中的第 i 个特征的最大最小值分别为 $max\{x_i\}$ 和 $min\{x_i\}$, 则通过式 (5) 的归一化处理将训练和测试集中的样本特征值映射到 $[0, 1]$ 区间:

$$x'_i = \frac{x_i - \min\{x_i\}}{\max\{x_i\} - \min\{x_i\}} \quad (5)$$

从这些数据集中选出 10 个用于构建元学习样本库, 5 个作为测试的目标数据集。在构建元学习库时, 为了增加样本库的数量, 采用集成学习中扰动的思想, 对这 10 个数据集进行进一步扰动: 对特征的扰动, 例如从数据集中随机选出一些特征进行重新构建数据集; 对样本个数进行扰动, 缩放样本集大小等方式, 建立一个样本数为 100 的元数据集。根据该数据集进行回归训练获得最终的元回归模型。

对以上经过预处理的样本数据集, 分别使用文献 [4] 提出的 ReliefF 方法、文献 [6] 提出的 BPSO-SVM 方法, 文献 [7] 的 SVM-RFE 以及本文的 ML-BPSO 方法进行特征选择, 并对测试样本进行分类, 分别比较选择的时间、得到的特征维数、测试的诊断精度。其中, BPSO 的种群规模取 20, 最大迭代次数取 100, SVM 的训练工具采用 LIBSVM^[14]。得到实验结果如表 2 所示。

表 1 实验数据集

数据集		数据集规模	特征数	类别数
元学习 样本库 数据集	Pen Recognition	10 992	16	10
	Hypothyroid	3 772	29	4

	Molecular splice	3 190	60	3
测试数 数据集	Multi-feature	2 000	216	10
	Lymphography	148	18	4
	Cardiotocography	2 126	22	3
	Molecular splice	3 190	60	3
	Arrhythmia	452	279	16
Isolet	1 559	618	26	

表 2 实验结果

实验结果		特征维数	分类精度/(%)	时间比
Lymphography (79.05%)	ReliefF	5	75.82	0.75
	BPSO-SVM	6	79.27	34.87
	SVM-RFE	5	79.25	2.34
	ML-BPSO	6	79.27	1.00
Cardiotocography (98.44%)	ReliefF	4	98.50	0.24
	BPSO-SVM	3	98.73	128.52
	SVM-RFE	5	98.68	4.87
Molecular splice (92.45%)	ML-BPSO	3	98.73	1.00
	ReliefF	12	91.39	0.18
	BPSO-SVM	7	93.98	147.88
Arrhythmia (68.80%)	SVM-RFE	7	93.98	8.36
	ML-BPSO	7	93.98	1.00
	ReliefF	29	71.20	0.63
Isolet (93.97%)	BPSO-SVM	16	75.24	48.42
	SVM-RFE	23	73.45	6.58
	ML-BPSO	18	75.17	1.00
ReliefF	ReliefF	127	82.69	0.38
	BPSO-SVM	106	94.21	140.69
	SVM-RFE	148	89.63	4.17
ML-BPSO	113	93.52	1.00	

其中, 时间比是其它方法同本文方法所用时间的比值, 括号中的数字为原数据集的经过 SVM 的分类精度。从实验结果可以发现基于 Filter 思想的 ReliefF 特征选择方法能够获得最低的选择时间。这是因为在 Filter 的选择中仅分析数据之间的关系, 不考虑分类, 但是它获得的分类精度也低于其它方法。BPSO-SVM 方法选择出来的特征得到了最高的分类精度, 但是它所花的时间高于其它方法, 这主要是因为它每次迭代过程中均需要进行 SVM 训练。SVM-RFE 中 SVM 训练次数较 BPSO-SVM 大大减少, 判断各个特征的贡献率, 因此它的寻优时间较短。本文提出的 ML-BPSO 算法能够在较短的寻优时间获得与 BPSO-SVM 相当的分类精度。从时间上看, 算法仅略慢于 ReliefF, 这是由于在计算过程中分类精度采用元方法估算, 运算量主要集中在元特征的提取上, 而这部分计算远远低于 SVM 的训练。从选择的结果看, 最终结果分类精度与 BPSO-SVM 相当, 说明 ML-BPSO 算法能够很好的进行特征选择。

3 基于 ML-BPSO 的网络故障特征选择

由于目前网络中各种攻击事件和病毒越来越多，导致网络中产生大量的“软故障”，如网络服务异常、操作系统崩溃、链路拥塞甚至中断，因此利用攻击下的网络状态进行故障模拟具有可行性。本文选择了 DARPA 入侵数据集^[13]，以攻击下的网络状态模拟网络故障。该数据集包含四类网络攻击，分别是 Dos、Probe、R2L 和 U2R，用一套标准格式的数据来记录各种网络状态下的特征，每条记录均有 41 个特征值。按照各值的内容分为三组：基本特征集（Basic features）、内容特征集（Content features）和流量特征集（Traffic features）。为了确保数据的普适性，从原始数据集中以等间隔采集法选取训练集样本和测试样本，具体情况如表 3 所示。

表 3 实验样本集结构

类别	训练样本集	测试样本集
Normal	700 条	1 800 条
Dos	500 条	1 480 条
R2L	400 条	1 460 条
Probe	500 条	1 450 条
U2R	20 条	25 条

同样的，我们对数据集进行归一化处理，分别采用 ReliefF, BPSO-SVM, SVM-RFE 以及本文的 ML-BPSO 算法进行特征选择，粒子群算法种群规模为 20，最大迭代次数 100。实验结果如表 4 所示。

表 4 故障特征选择结果

算法	维数	特征维数编号	诊断精度	时间比
原始数据	41	——	92.07	——
ReliefF	16	5,7,8,9,17,24,25,26,27,28,29,31,35,37,38,39	92.91	0.43
BPSO-SVM	8	6,11,14,17,24,29,34,38	93.95	148.35
SVM-RFE	13	3,5,7,11,17,25,27,28,31,32,37,38,39	90.94	5.28
ML-BPSO	10	3,5,6,7,11,17,22,24,32,33	93.82	1

从实验结果可以看出同样的 Wrapper 算法，ML-BPSO 的运行时间远远低于 BPSO-SVM，而且最后的诊断精度与 BPSO-SVM 相当。说明本文算法能够在不损失诊断精度的条件下大幅度的降低传统采用分类训练来计算分类精度的特征选择方法。当然，元模型的构建过程需要进行大量的运算，我们在元模型建立过程中相当于使用了 100 个样本集，加上 5 折交叉验证，需要 500 次 SVM 训练。然而在一次种群规模取 20，最大迭代次数取 100 的 Wrapper 型特征选择过程中需要 10000 次的 SVM 训练，可见本文方法可以大大降低运算量。同时，当元模型建立后，我们对其他多个样本集进行特征选择时均可使用这一模型。

4 结束语

特征选择在网络故障诊断领域能够提高诊断的精度和速

度，作为目前特征提取效果较好的 Wrapper 模式选择算法由于计算目标函数时需要进行分类训练，带来大量的计算。本文从降低分类器的精度确定过程的运算量出发，通过元学习方法估算分类器的精度避免训练过程。并对分类结果和选择出的样本维数的综合考虑，利用 BPSO 在特征空间中进行全局搜索选出最优特征集。最后，在 DARPA 数据集上的网络故障特征选择实验表明本文提出的方法能够快速选择出最优故障特征，能够为网络故障诊断中的特征选择问题提供一种新的途径。

参考文献:

- [1] Zhu Z, Ong Y, Dash M. Wrapper-filter feature selection algorithm using a memetic framework [J]. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 2007, 37 (1): 70-76.
- [2] Zhao M Y, Fu C, Ji L P, et al. Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes [J]. Expert Systems with Applications 2011, 38 (5): 5197-5204.
- [3] Kim S, Oommen B. On using prototype reduction scheme to optimize kernel-based Fisher discriminant analysis [J]. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 2008, 38 (2): 564-570.
- [4] Kononenko I. Estimating attributes: Analysis and extensions of RELIEF [J]. Machine Learning. 1994, 784: 171-182.
- [5] Zhang L, Meng X R, Wu W J, et al. Network Fault Feature Selection Based on Adaptive Immune Clonal Selection Algorithm [A]. 2009 International Joint Conference on Computation Sciences and Optimization [C]. Hainan, China; 2009, 969-973.
- [6] 潘 泓, 李晓兵, 金立左, 等. 一种基于二值粒子群优化和支持向量机的目标检测算法 [J]. 电子与信息学报, 2011, 33 (1): 117-121.
- [7] Guyon I, Weston J, Barnhill S, et al. Gene Selection for Cancer Classification Using Support Vector Machines [J]. Machine Learning, 2002, 46 (1/3): 389-422.
- [8] Taciana A F, Gomes Ricardo B C, Prudencio, et al. Combining meta-learning and search techniques to select parameters for support vector machines [J]. Neurocomputing, 2012, 75 (1): 3-13.
- [9] Munoz M, Kirley M, Halgamuge S. A Meta-learning Prediction Model of Algorithm Performance for Continuous Optimization Problems [A]. LNCS [C]. 7491, Springer, Berlin, Heidelberg, 2012: 226-235.
- [10] Guo X, Yang J, Wu C, et al. A novel LS-SVMS hyper-parameter selection based on particle swarm optimization [J]. Neurocomputing, 2008, 71 (16-18): 3211-3215.
- [11] Sotoca J M, Mollineda R A, Sanchez J S. A meta-learning framework for pattern classification by means of data complexity measures [A]. Intelligence Artificial, Revista Ibero americana de IA [C]. 2006, 29 (10): 31-38.
- [12] Kennedy J, Eberhart R C, Shi Y H. Swarm intelligence [M]. 北京: 人民邮电出版社, 2009.
- [13] University of California Irvine. UCI KDD Archive [DB/OL]. http://kdd.ics.uci.edu/
- [14] Chang C C, Lin C J. LIBSVM: A library for support vector machines [EB/OL]. http://www.csie.ntu.edu.tw/~cjlin/libsvm.