
算法 1 特征抽取

输入: *data* 时间序列对应的数据, *label* 时间序列对应的标签, *window_len* 窗口长度

输出: 时间序列抽取的特征矩阵和标签

```

1: function EXTRACT_FEATURES(data, label, window_len)
2:   features  $\leftarrow$  0
3:   targets  $\leftarrow$  0
4:   i  $\leftarrow$  0
5:   num_windows  $\leftarrow$  len(data)/(window_len/2)
6:   for i = 0  $\rightarrow$  num_windows do
7:     target  $\leftarrow$  int(label[i  $\rightarrow$  i + window_len].mode())
8:     targets  $\leftarrow$  targets  $\cup$  target
9:     for c = 0  $\rightarrow$  data.columns do
10:      features  $\leftarrow$  features  $\cup$  FEATURIZE(data[i  $\rightarrow$  i + window_len])
11:    end for
12:    i  $\leftarrow$  i + window_len/2
13:  end for
14:  return features, targets
15: end function

```

$$Precision = \frac{TP}{(TP + FP)}$$

$$Recall = \frac{TP}{(TP + FN)}$$

$$F1 - Score = \frac{2 * Precision * Recall}{(Recall + Precision)}$$

$$Accuracy = \frac{TP}{(TP + FP + FN + TN)}$$

算法 2 FEATURIZE

- 1: $rms_val = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}$
 - 2: $min_max_mean = \frac{1}{len} \sum_{i=1}^{len} |max_i - min_i|$ 其中 $len = \min(max, min)$; min 和 max 为序列中的极大值和极小值点
 - 3: $peak = max\ max - min\ min$
 - 4: $peaknum = len(max) + len(min)$
 - 5: $mean = \frac{1}{n} \sum_{i=1}^n x_i$
 - 6: $standarddeviation = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i - mean}$
 - 7: $coefficientsofvariation = \frac{standarddeviation}{mean}$
 - 8: $Skewness = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - mean)^3}{(\frac{1}{n} \sum_{i=1}^n (x_i - mean)^2)^{\frac{3}{2}}}$
 - 9: $Kurtosis = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - mean)^4}{(\frac{1}{n} \sum_{i=1}^n (x_i - mean)^2)^2} - 3$
 - 10: $log - energy = \frac{1}{n} \sum_{i=1}^n \log(x_i^2)$
-

算法 3 单一活动类别识别

- 1: **function** TASK1
 - 2: $features \leftarrow 0$
 - 3: $targets \leftarrow 0$
 - 4: **for** $i = 0 \rightarrow 12$ **do**
 - 5: $data \leftarrow data \cup PREPROCESSING(i)$
 - 6: **for** $i = 1 \rightarrow 7$ **do**
 - 7: $d \leftarrow data[label = i]$
 - 8: $feature, target = EXTRACT_FEATURES(d[x, y...], d[label], 520)$
 - 9: $features \leftarrow features \cup feature$
 - 10: $targets \leftarrow targets \cup target$
 - 11: **end for**
 - 12: **end for**
 - 13: $classifiers = Train(features, targets)$
 - 14: **for** $i = 13 \rightarrow 15$ **do**
 - 15: $data \leftarrow data \cup PREPROCESSING(i)$
 - 16: **for** $i = 1 \rightarrow 7$ **do**
 - 17: $feature, target = EXTRACT_FEATURES(data[label = i][x, y...], data[label = i][label], 520)$
 - 18: TEST(classifiers, feature, target)
 - 19: **end for**
 - 20: **end for**
 - 21: **end function**
-

算法 4 多活动类别识别

```

1: function TASK3
2:    $features \leftarrow 0$ 
3:    $targets \leftarrow 0$ 
4:   for  $i = 0 \rightarrow 12$  do
5:      $data \leftarrow data \cup \text{PREPROCESSING}(i)$ 
6:      $feature, target = \text{EXTRACT\_FEATURES}(data[x, y...], data[label], 3)$ 
7:      $features \leftarrow features \cup feature$ 
8:      $targets \leftarrow targets \cup target$ 
9:   end for
10:   $classifiers = \text{Train}(features, targets)$ 
11:  for  $i = 13 \rightarrow 15$  do
12:     $data \leftarrow data \cup \text{PREPROCESSING}(i)$ 
13:     $feature, target = \text{EXTRACT\_FEATURES}(data[x, y...], data[label], 3)$ 
14:     $\text{TEST}'(classifiers, feature, target)$ 
15:  end for
16: end function

```

算法 5 多活动类别识别

```

1: function TASK3
2:    $features \leftarrow 0$ 
3:    $targets \leftarrow 0$ 
4:   for  $i = 0 \rightarrow 12$  do
5:      $data \leftarrow data \cup \text{PREPROCESSING}(i)$ 
6:      $feature, target = \text{EXTRACT\_FEATURES}(data[x, y...], data[label], 3)$ 
7:      $features \leftarrow features \cup feature$ 
8:      $targets \leftarrow targets \cup target$ 
9:   end for
10:   $classifiers = \text{Train}(features, targets)$ 
11:  for  $i = 13 \rightarrow 15$  do
12:     $data \leftarrow data \cup \text{PREPROCESSING}(i)$ 
13:     $feature, target = \text{EXTRACT\_FEATURES}(data[x, y...], data[label], 3)$ 
14:     $\text{TEST}'(classifiers, feature, target)$ 
15:  end for
16: end function

```

算法 6 梯度下降法

输入: $data$ 数据矩阵, $label$ 标签

输出: 权重

```
1: function GRADASCENT( $data, label$ )
2:    $\alpha \leftarrow 0.001$ 
3:    $maxCycles \leftarrow 500$ 
4:    $weights \leftarrow ones((n, 1))$ 
5:   for  $k = 0 \rightarrow maxCycles$  do
6:      $h \leftarrow SIGMOID(data * weights)$ 
7:      $error \leftarrow (label - h)$ 
8:      $weights \leftarrow weights + \alpha * data * error$ 
9:   end for
10:  return  $weights$ 
11: end function
```

算法 7 决策树

输入: $data$ 数据矩阵, $Target_attribute$ 要预测的目标属性, $Attributes$ 除目标属性外供学习到的决策树测试属性列表

输出: $root$ 一颗能正确分类给定 $data$ 的决策树

```

1: function DECISIONTREECLASSIFIER( $data, label$ )
2:   创建  $root$  节点
3:   if 所有  $data$  都为正 then
4:     return  $label = +$  的单节点树  $root$ 
5:   end if
6:   if 所有  $data$  都为负 then
7:     return  $label = -$  的单节点树  $root$ 
8:   end if
9:   if  $attributes$  为空 then
10:    return 单节点树  $root$ ,  $label = data$  中最普遍的  $Target\_attributes$ 
11:  end if
12:   $A \leftarrow attributes$  中分类  $data$  能力最好的属性
13:   $root$  的决策属性  $\leftarrow A$ 
14:  for  $i \rightarrow len(A)$  do
15:     $a \leftarrow A[i]$ 
16:    在  $root$  下加一个新的分支对应测试  $A=a$ 
17:    令  $data(a)$  为  $data$  中满足  $A$  属性值为  $a$  的子集
18:    在这个新分支下一个子树 DECISIONTREECLASSIFIER( $data(a), Target\_attribute, Attributes - A$ )
19:  end for
20:  return  $root$ 
21: end function

```
