

---

**Algorithm 1** TAN算法

---

输入: *train\_data*训练集, *test\_data*测试集, *attribute\_list*属性对象列表

输出: 分类正确率

```

1: function TAN(train_data, test_data, attribute_list)
2:    $y1 \leftarrow \text{Class1\_Label}, y2 \leftarrow \text{Class2\_Label};$  ▷ 获取类标签
3:   for each attribute do ▷ 计算不考虑属性依赖的后验概率
4:      $P(\text{attribute}|y1), P(\text{attribute}|y2) \leftarrow \text{computeNB}(\text{train\_data});$ 
5:   end for
6:    $\text{weights} \leftarrow \text{computeWeight}(\text{train\_data});$  ▷ 计算属性之间的权重
7:    $\text{parent\_list} \leftarrow \text{Prim}(\text{weights});$  ▷ 采用Prim算法构造最大边权生成树所得到的被依赖属性
8:   for each attribute and parent_attribute do ▷ 计算考虑属性两两依赖的后验概率
9:      $P(\text{attribute}|\text{parent\_attribute}, y1), P(\text{attribute}|\text{parent\_attribute}, y2) \leftarrow \text{computeTAN}(\text{train\_data});$ 
10:  end for
11:   $\text{correctClassified}, \text{incorrectlyClassified} \leftarrow 0;$  ▷ 初始化正确分类和错误分类实例数量
12:  for instance in test_data do ▷ 对于测试数据集中每个实例
13:     $\text{numerator} \leftarrow P(y1), \text{denominator} \leftarrow P(y2);$ 
14:     $\text{index} \leftarrow 0;$ 
15:    for cur_attr_value in instance do ▷ 对于实例中每个属性
16:      if  $\text{index} == \text{max}$  then
17:        break;
18:      end if
19:      if  $\text{parent\_list.contains}(\text{attribute\_list}[\text{index}].\text{attributeName})$  then ▷ 若依赖某个属性
20:         $\text{parent} \leftarrow \text{parent\_list}[\text{attribute\_list}[\text{index}].\text{attributeName}];$ 
21:         $\text{parent\_value} \leftarrow \text{instance}[\text{parent.index}];$ 
22:         $\text{numerator} \leftarrow \text{numerator} * P(\text{cur\_attr\_value}|\text{parent\_value}, y1);$ 
23:         $\text{denominator} \leftarrow \text{denominator} * P(\text{cur\_attr\_value}|\text{parent\_value}, y2);$ 
24:      else ▷ 否则
25:         $\text{numerator} \leftarrow \text{numerator} * P(\text{cur\_attr\_value}|y1);$ 
26:         $\text{denominator} \leftarrow \text{denominator} * P(\text{cur\_attr\_value}|y2);$ 
27:      end if
28:       $\text{index} \leftarrow \text{index} + 1;$ 
29:    end for
30:     $P(y1|\text{instance}) \leftarrow \text{numerator} / (\text{numerator} + \text{denominator});$  ▷ 验证
31:     $P(y2|\text{instance}) \leftarrow \text{denominator} / (\text{numerator} + \text{denominator});$ 
32:    if  $P(y1|\text{instance}) \geq P(y2|\text{instance})$  then ▷ 分类为y1
33:      if  $y1 == \text{instance.ClassLabel}$  then ▷ 分类正确
34:         $\text{correctClassified} \leftarrow \text{correctClassified} + 1;$ 
35:      else ▷ 分类错误

```

```

36:         incorrectlyClassified  $\leftarrow$  incorrectlyClassified + 1;
37:     end if
38: else ▷ 分类为 $y_2$ 
39:     if  $y_2 == \textit{instance.ClassLabel}$  then
40:         correctClassified  $\leftarrow$  correctClassified + 1;
41:     else
42:         incorrectlyClassified  $\leftarrow$  incorrectlyClassified + 1;
43:     end if
44: end if
45: end for
46: return correctClassified / (correctClassified + incorrectlyClassified) ▷ 分类正确率
47: end function

```

---