LaTeX 中文报告模板

姓名

2024 年 4 月 17 日

目录

第一章 LaTeX

1.1 文本

1. 正常
2. **加粗**
3. *斜体*
4. 下划线
5. highlight

1.2 图片

1.2.1 单图

图 1.1: SCUT 的 Logo

1.2.2 子图

(a) 子图 1

(b) 子图 2

(c) 子图 3

(d) 子图 4

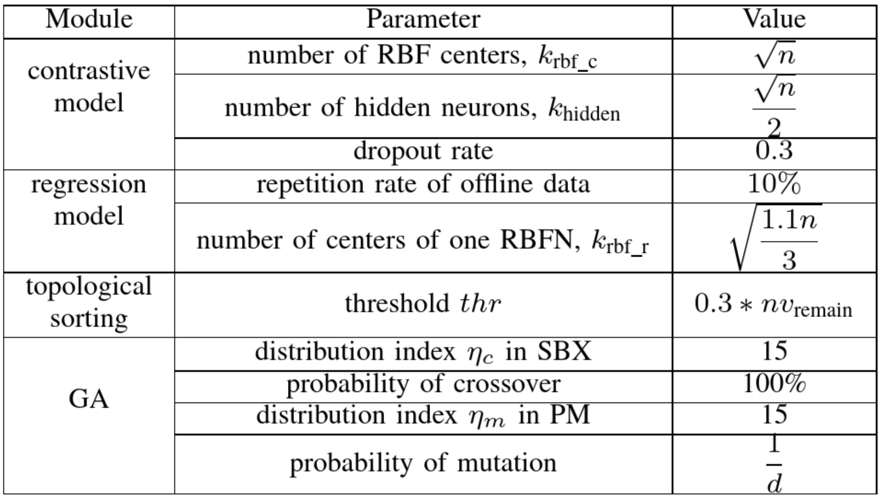
图 1.2: 子图

1.3 表

表 1.1: 参数值

Parameter	Value
α	1
β	1

表 1.2: 参数值



Module	Parameter	Value
contrastive model	number of RBF centers, $k_{\text{rbf\_c}}$	$\sqrt{n}$
	number of hidden neurons, k_{hidden}	$\frac{\sqrt{n}}{2}$
	dropout rate	0.3
regression model	repetition rate of offline data	10%
	number of centers of one RBFN, $k_{\text{rbf\_r}}$	$\sqrt{\frac{1.1n}{3}}$
topological sorting	threshold thr	$0.3 * nv_{\text{remain}}$
GA	distribution index η_c in SBX	15
	probability of crossover	100%
	distribution index η_m in PM	15
	probability of mutation	$\frac{1}{d}$

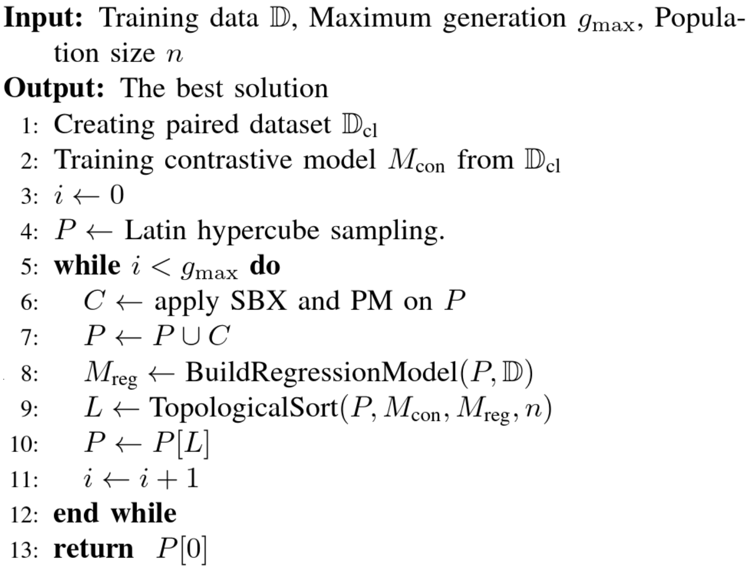
有时候太懒了，直接截图，把图片扔到 table 环境，例如上边的表。

1.4 伪代码

Algorithm 1 KahnAlgorithm

```
Input: Graph $G(\mathbb{V}, \mathbb{E})$
Output: Sequence $L$
$L \leftarrow$ an empty sequence
$Q \leftarrow$ the vertices whose indegree is zero
while $Q$ is not empty do
    $u \leftarrow$ remove the top node of $Q$
    add $u$ to $L$
    for each node $v$ with an edge $e$ from $u$ to $v$ do
        remove edge $e$ from graph $G$
        if indegree of $v$ is 0 then
            push $v$ to $Q$
        end if
    end for
end while
return $L$
```

Algorithm 2 框架



```
Input: Training data $\mathbb{D}$, Maximum generation $g_{\max}$, Population size $n$
Output: The best solution
1: Creating paired dataset $\mathbb{D}_{\text{cl}}$
2: Training contrastive model $M_{\text{con}}$ from $\mathbb{D}_{\text{cl}}$
3: $i \leftarrow 0$
4: $P \leftarrow$ Latin hypercube sampling.
5: while $i < g_{\max}$ do
6:     $C \leftarrow$ apply SBX and PM on $P$
7:     $P \leftarrow P \cup C$
8:     $M_{\text{reg}} \leftarrow \text{BuildRegressionModel}(P, \mathbb{D})$
9:     $L \leftarrow \text{TopologicalSort}(P, M_{\text{con}}, M_{\text{reg}}, n)$
10:    $P \leftarrow P[L]$
11:    $i \leftarrow i + 1$
12: end while
13: return $P[0]$
```

有时候太懒了，直接截图，把图片扔到 algorithm 环境，例如上边的算法。

1.5 高亮

```cpp
#include <algorithm>
using namespace std;
void quickSort(int arr[],
               int begin,
               int end) {
  int i, j, t, pivot;
  if (begin > end)
    return;

  pivot = arr[begin];
  i = begin;
  j = end;
  while (i != j) {
    while (arr[j] >= pivot && i < j)
      j--;
    while (arr[i] <= pivot && i < j)
      i++;
    if (i < j)
      swap(arr[i], arr[j]);
  }

  arr[begin] = arr[i];
  arr[i] = pivot;
  quickSort(arr, begin, i - 1);
  quickSort(arr, i + 1, end);
}
```

1.6　多栏

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

1.7　数学

Interline Formula:

$$a_n = a_{n-1} + 1 \tag{1.1}$$

行内公式: 这是一个简单的等差数列公式 $a_n = a_{n-1} + 1$ 。

1.8　引用

- 图片: 图 1.1
- 子图: 子图 1.2a
- 表: 表 1.1
- 伪代码: 算法 1
- 公式: 式 1.1
- 章: chapter 一
- 论文:[1]
- URL 1: baidu
- URL 2: https://baidu.com

参考文献

[1] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.